

IoT Hackathon -

Eine Ardublock Blaupause für eigene Ideen



Plattform Innovative Digitalisierung der Wirtschaft
Fokusgruppe Intelligente Vernetzung

Nationaler IT Gipfel

1	IoT Hackathon IT-Gipfel 2016	3
1.1	Entwicklungsumgebung	5
1.1.1	Arduino-Plattform	5
1.1.2	IoT-Entwicklungs-kit (ESP-8266).....	5
1.1.3	Interfacekomponenten (Library) einbinden	8
1.1.4	Grafische Programmieroberfläche Ardublock.....	9
1.2	Blaupausen für eigene Projekte	11
1.2.1	Hallo Arduino - Hallo IoT-Kit.....	14
1.2.2	Die digitale Flaschenpost – Hallo Server.....	20
1.2.3	Unser Forschungsschiff – Hallo IoT	23
1.2.4	Forschungsschiff informiert sich – Hallo M2M.....	28
1.2.5	Forschungsschiff greift ein - Hallo Control	29
1.2.6	Wearables – Forschungsschiff kleidet sich.....	34
1.2.7	Die Flaschenpost wird flexibel – Hallo Interface....	36
1.3	Für Hacker & Maker	39
1.3.1	Schaltplan	39

Vorwort

Wie gelingt der digitale Wandel unserer Gesellschaft? Welche Auswirkungen hat das Internet der Dinge auf die Wirtschaft? Wie hält die schulische Ausbildung mit den Anforderungen der modernen Arbeitswelt Schritt?

Diese und ähnliche Fragen thematisiert der jährlich stattfindende Nationale IT-Gipfel der Bundesregierung. Die Spitzenkräfte aus Politik, Wirtschaft, Wissenschaft und Zivilgesellschaft treffen sich in 2016/17, um sich zum Leitthema „Digitale Bildung“ auszutauschen und Handlungsempfehlungen zu diskutieren.

Gemeinsam mit rund 200 Experten aus mehr als 150 Unternehmen und Institutionen bringt die Fokusgruppe „Intelligente Vernetzung“ (FG2) des IT-Gipfels die digitale Vernetzung Deutschlands voran. Ein Ziel: Erste Programmierkenntnisse schon in der Schule vermitteln. Unter dem Motto „Das Internet der Dinge (IoT) anfassbar machen“ plant die Expertengruppe die Austragung eines „Hackathon“ für Schüler.

Schülergruppen aus Rheinland-Pfalz und dem Saarland treffen sich hierzu am Vortag des Gipfels am Umwelt-Campus in Birkenfeld und der HTW in Saarbrücken, um gemeinsam Ideen zum Thema zu entwickeln und mit Hilfe eines von der Expertengruppe bereitgestellten Baukastens auch gleich vor Ort umzusetzen. Die erfolgreichsten Gruppen stellen ihre Ergebnisse dann am nächsten Tag auf dem Nationalen IT-Gipfel vor.

Natürlich kann ein solches singuläres Ereignis keine systematische Bildungsstrategie ersetzen, es soll aber zeigen, dass die technischen Voraussetzungen keine unüberwindbare Hürde für Schulen darstellen und dass das Thema eine nachhaltige intrinsische Motivation der Schüler auslöst. Das Thema Internet der Dinge macht einfach Spaß und Lust auf mehr.

Wer sofort einsteigen möchte startet mit Kapitel 11.

1 IoT Hackathon - eine Blaupause für eigene Ideen

Autoren: Klaus Gollmer¹, Guido Burger²

Die digitale Transformation wird unsere Wirtschaft und Gesellschaft in den nächsten Jahrzehnten grundlegend verändern. Der selbstverständliche Umgang mit Sensoren und Kommunikationsmodulen, aber auch deren Programmierung bis hin zur Cloud-Anwendung ist eine Voraussetzung für neue Anwendungsideen und Geschäftsmodelle.

Das dazu notwendige **algorithmische Denken** sollte deshalb zukünftig schon in der Schule geübt werden. Hierzu stellen wir beispielhafte Ideen und ihre technische Umsetzung vor. Fokussiert auf ein schnelles Erfolgserlebnis bieten die kleinen Programme einen Kristallisationspunkt für die Beschäftigung mit dem Thema und der kreativen Umsetzung eigener Ideen.

Im Wettbewerb mit mehreren Teams (**Hackathon**) macht es noch viel mehr Spaß: es entstehen konkurrierende Ideen, aber vielleicht auch ganz neue Lösungen, da man mit anderen Teams zusammenarbeiten kann. Durch die vorbereiteten Beispiele gelingt der Einstieg kinderleicht. Ein für speziell für den Hackathon entwickeltes IoT-Board in Verbindung mit einer angepassten grafischen Programmieroberfläche, sowie den Grove Interfacekomponenten erleichtern die praktische Umsetzung auch ohne Programmierkenntnisse.

Um den unterschiedlichen Kenntnisstand der Leser zu berücksichtigen haben wir uns für zwei Versionen der Blaupausen entschieden: Grafische Programmierung mit **Ardublock** bzw. die direkte Programmierung in der **Hochsprache C**.

¹ www.umwelt-campus.de

² www.fab-lab.eu

4 Das Internet der Dinge anfassbar machen

Durch einfache Modifikation der Beispiele entstehen daraus komplexere Lösungen für die kleinen Probleme des Alltags:

- **Umwelt:** Sorge dafür, dass die Bewohner von A-Bach und B-Tal in Zukunft rechtzeitig vor einem folgenschweren Unwetter gewarnt werden. Entwerfe einen geeigneten Umweltsensor (z. B. zur Messung der Bodenfeuchtigkeit am Berghang, des Pegels des nächsten Flusses usw.) und verknüpfe die Sensordaten mit dem Wetterbericht.
- **Verkehr:** Reduziere das Verkehrsaufkommen und die Umweltbelastung durch Schadstoffe. Entwickle eine IoT-Anwendung, damit eine Mülltonne vor der Müllabfuhr nur dann angefahren wird, wenn sie voll ist.
- **Industrie 4.0:** Hilf mit, die Arbeitsplätze der Fabrik AG zu sichern. Erfinde intelligente Sensoren (z.B. Stromverbrauch, Bewegung), um die Effizienz der Produktionsmaschinen und somit die Wettbewerbsfähigkeit der gesamten Firma zu verbessern.
- **Kleidung:** Nutze Sensoren und verleihe Kleidungsstücken, zum Beispiel über innovative Lichtelemente, einen neuen Ausdruck. Dabei können Daten über die Bewegung eines Menschen erfasst werden (Quantified Self). Dadurch entstehen neue kreative Anwendungen für Freizeit und Sport sowie für ältere Menschen.
- **Virtuelle Welten:** Verknüpfe die virtuelle und die reale Welt unter Nutzung einer Spiele-Plattform im Internet. Spielerisch gelingt der Einstieg in die Sensorik und Programmierung.

Das System ist als **digitale Flaschenpost** konzipiert und trägt damit als Metapher die Nachricht des einfachen Internetzugangs für die Dinge dieser Welt in die Politik und an die Schulen und Hochschulen. In einem zweiten Schritt schicken wir die Flasche als **Forschungsschiff** auf die Reise, um unsere Umwelt zu erkunden und neue digitale Welten zu erobern. So verbindet sich das Internet mit der analogen Welt in der wir leben.

1.1 Entwicklungsumgebung

Die folgende Kurzeinführung soll den Zugang zu Werkzeugen, Technologie und Kreativitätsmethoden erleichtern. Zuerst wird die Arduino-Entwicklungsumgebung und die Hardware vorgestellt, um anschließend anhand einfacher Beispiele in die Welt des **Internets der Dinge** (IoT) einzudringen.

1.1.1 Arduino-Plattform

Mit der Open-Source Arduino-Entwicklungsumgebung steht eine in vielen Projekten bewährte Plattformtechnologie zur freien Verfügung. Softwareseitig bildet der GNU-C-Compiler den Kern einer einfach zu bedienenden Entwicklungsoberfläche. Auf dem Hackathon ist die Entwicklungsumgebung bereits installiert. Eine Installationsanleitung für den heimischen Rechner findet sich auf der Arduino-Homepage³.

1.1.2 IoT-Entwicklungskit (ESP-8266)

Hardwareseitig sind unzählige kompatible Entwicklungsboards verschiedener Hersteller für wenig Geld verfügbar. Seit kurzem existiert mit dem ESP-8266 eine Plattform⁴, die sich aufgrund des niedrigen Preises und der WLAN-Konnektivität ideal für Ausbildungszwecke eignet. Kommerziell erhältliche Entwicklungsboards, wie NodeMCU⁵ oder Wemos D1 mini⁶, enthalten alle für die Programmentwicklung und Ausführung notwendigen Komponenten inklusive USB und WLAN. Je nach Anwendungsfall werden die benötigten Ein- /Ausgabekomponenten, wie LED, Taster, Sensoren und Aktoren manuell hinzugefügt – ein im schulischen Alltag sehr fehleranfälliges Unterfangen.

³ arduino.cc/en/Main/Software

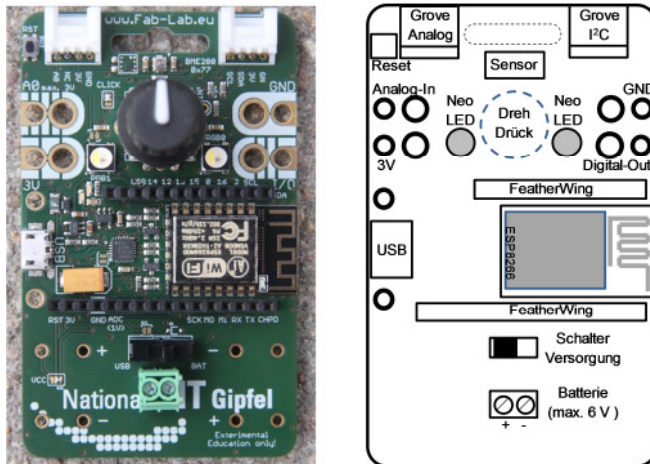
⁴ mikrocontroller.net/articles/ESP8266

⁵ nodemcu.com

⁶ Wemos.cc

6 Das Internet der Dinge anfassbar machen

Hier reduziert das von uns speziell für Schüler entwickelte IoT-Kit die Verdrahtungsproblematik durch eine Reihe bereits auf dem Board integrierter Komponenten.



IoT- Abb. 1: Das Experimentier-Kit

Das **IoT-Experimentier-Kit** legt damit als quelloffenes Projekt (Schaltplan im letzten Kapitel) die Blaupause für die Ausbildung an Schulen und Hochschulen. Produziert und entwickelt in Deutschland folgt es folgenden Prinzipien:

- **Kein Löten:** Erweiterungen werden einfach angesteckt, und sind durch eine Vielzahl von Sensoren und Aktoren flexibel an die eigene Problemstellung anpassbar.
- **Sensorik:** ausgestattet mit einem Bosch-Sensortec Umweltsensor (Barometer, Luftfeuchtigkeit, Temperatur) wird das Internet der Dinge in wenigen Minuten erlebbar.
- **Interaktiv:** durch „Dreh/Drück“ Eingabe und unterschiedliche Anzeigen (farbige NeoPixel, einfarbige LED).
- **Flexible Energieversorgung:** über USB-Schnittstelle oder unter Verwendung von Standard Batterien bzw. Akkus. Das Laden via Solarstrom ist als „hack“ vorgesehen.
- **Kompatibel:** zur Schulausstattung (Bananenstecker, WLAN und USB) und vorhandenen Communities mit vielen Anwendungsbeispielen (Arduino, Adafruit, SeeedStudio Grove). Die Entwicklungsumgebung läuft auf Windows, Linux und MacOS.

Das Internet der Dinge anfassbar machen

Im Rahmen des Hackathon ist das Board ohne weitere Konfiguration direkt einsetzbar. Da es sich um ein sogenanntes „Third Party Board“ handelt, erfolgt die Anbindung an die Entwicklungsumgebung über den „Boardverwalter“. Um das Board auch am häuslichen PC nutzen zu können muss dort zuerst unter „Datei“ im Untermenü „Voreinstellungen“ die folgende Boardverwalter-URL eingetragen werden:

„http://arduino.esp8266.com/stable/package_esp8266com_index.json“.

Anschließend im Menü „Werkzeuge“ im Untermenü „Board“ den „Boardverwalter“ anwählen und die esp8266 Toolchain installieren. Nach erfolgreicher Installation steht schließlich das Board zur Auswahl zur Verfügung. Dazu unter Werkzeuge, das kompatible Board „NodeMCU V1.0 (ESP12E Modul)“ anwählen. Nach Anschluss des Kits an den USB-Port des PCs sollte der entsprechende Treiber vom Betriebssystem automatisch initialisiert werden. Bei Problemen kann der Treiber manuell nachinstalliert werden⁷

Anschließend unter „Werkzeuge“ -> „Port“ den vom Betriebssystem vergebenen COM-Port eintragen.

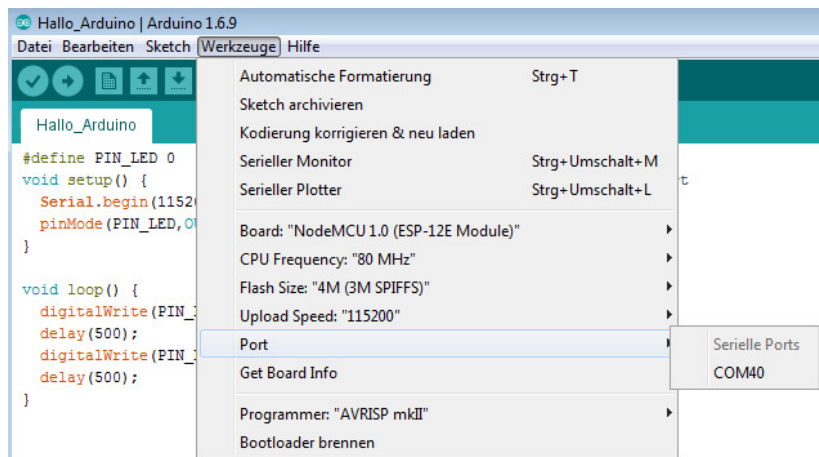


Abb. 2: Einbindung in die Arduino-Entwicklungsumgebung

⁷ <https://www.silabs.com/products/mcu/Pages/USBtoUARTBridgeVCPDrivers.aspx>

1.1.3 Interfacekomponenten (Library) einbinden

Die bereits auf dem Board integrierten Interfacekomponenten benötigen zur einfachen Ansteuerung die Einbindung spezieller Bibliotheken. Damit kann die betreffende Komponente dann im Programmcode sehr einfach über Arduino-Befehle angesprochen werden. Ein sonst übliches intensives Studium der Datenblätter entfällt. Dieses flexible Bibliothekskonzept ist eine der Gründe für die enorme Verbreitung der Arduino-Plattform. Insbesondere aufwendige Sensorprotokolle sind als Bibliotheken vorhanden und machen deren Nutzung komfortabel.

Auf dem Hackathon sind alle benötigten Bibliotheken bereits installiert. Diese können, ähnlich wie beim „Boardmanager“, mit Hilfe eines „Librarymanager“ eingebunden werden (Menü „Sketch“->„Bibliothek einbinden“). Um die Bibliotheken auch am häuslichen PC nutzen zu können, enthält die Tabelle alle im Rahmen unserer Blaupausen eingebundenen Komponenten.

Tab. 1: Einzubindende Bibliotheken

Interface-komponente	Wofür	Library
Bosch BME280	Temperatur, Druck, Feuchte	Sparkfun BME280 V1.0.0
NeoPixel	RGB-LED Anzeige	Adafruit NeoPixel V1.0.5
Dreh/Drück	Drehknopf (Rotary-Encoder)	Encoder by Paul Stoffregen V1.4.1
LED-Matrix	Anzeigematrix	Extern einzubinden ⁸ (Zip-Datei)
	Textausgabe	Adafruit GFX V1.1.5

⁸ learn.adafruit.com/adafruit-15x7-7x15-charlieplex-led-matrix-charliewing-featherwing/

1.1.4 Grafische Programmieroberfläche Ardublock

Insbesondere Programmierneulinge kämpfen ständig mit der Syntax der jeweiligen Programmiersprache. Die Schreibweise einzelner Befehle muss in der Regel mühselig im Handbuch nachgeschlagen werden und die Anordnung von Klammern, Blöcken, Semikolon usw. erscheint auf dem ersten Blick sehr verwirrend. Hier hilft der Einsatz einer grafischen Programmiersprache (wie z.B. Scratch⁹), bei der einzelne Programmblöcke per Mausklick zu einem Programm zusammengepuzzelt werden. Passende Schloss/Schlüssel-Mechanismen sorgen automatisch für ein syntaktisch korrektes Programm. Glücklicherweise gibt es auch für den Arduino eine entsprechende Erweiterung in Form der Oberfläche Ardublock¹⁰ bzw. der im Rahmen des Hackathon verwendeten IoT-Erweiterung¹¹ von Ardublock.

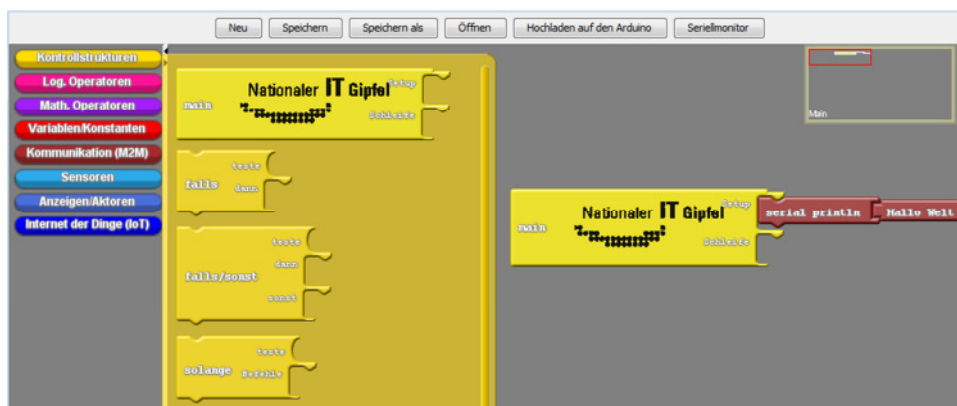


Abb. 3: Oberfläche Ardublock zu grafischen Programmierung

Hier existieren neben einfachen Blöcken zur Realisierung von Kontrollstrukturen, Variablen, logischen bzw. mathematischen Operationen auch spezifisch auf das IoT-Kit zugeschnittene

⁹ <https://scratch.mit.edu/>

¹⁰ <http://blog.ardublock.com/>

¹¹ <http://div-konferenz.de/events/hackathon/?view=1>

10 Das Internet der Dinge anfassbar machen

Module, die wir im Laufe der Blaupausen kennen lernen werden. Diese grafischen Blöcke besitzen spezifische Andockstellen (Schloss/Schlüssel), die sicherstellen, dass nur kompatible Blöcke zu einem Programm zusammengestellt werden können. Aus der grafischen Darstellung wird über den Menüpunkt „Hochladen auf den Arduino“ direkt der Hochsprachencode (C/C++) in Arduino-Syntax generiert und auf das Board geladen.

Zur Installation von Ardublock ist die Java Datei „ardublock-IoT.jar“ in den Unterordner „\tools\ArduBlockTool\tool\“ des Arduino Sketchbook Verzeichnisses zu kopieren. Das Sketchbook-Verzeichnis lässt sich in der Arduino-GUI unter „Datei“->“Voreinstellungen“ ermitteln. Nach Neustart der Arduino-GUI existiert fortan unter dem Punkt „Werkzeuge“ ein Eintrag „Ardublock“, über den die Oberfläche gestartet werden kann.

1.2 Blaupausen für eigene Projekte

Nach Einrichtung der Entwicklungsumgebung können wir nun mit der Umsetzung der Beispielprojekte beginnen. Wir starten mit einem einfachen „**Hallo Arduino**“ um uns mit der Bedienung der Entwicklungsplattform vertraut zu machen. Anschließend werden verschiedene Anwendungsszenarien erläutert:

- Im Projekt „**digitale Flaschenpost**“ senden wir eine Nachricht an unseren Flaschenpost-Server – hierdurch lernen wir die Anbindung an das Internet, nutzen eine WLAN Verbindung und eine einfache HTML Seite zur Eingabe der Nachricht, die Ausgabe der Nachricht nutzt eine Matrixanzeige.

Ziel: Wir lernen eine Kommunikationsstrecke aufzubauen, welche wir für unser späteres Forschungsschiff benötigen!

Mehr: Sicherheit ist für IoT wichtig, am Hackathon schauen wir bei den anderen Teams vorbei, welche IP Adresse nutzen die? Können wir Nachrichten an deren Flasche senden?

- Im Projekt „**Forschungsschiff**“ senden wir nun Daten von Sensoren in das Internet und visualisieren diese. Dabei nutzen wir wieder eine WLAN-Verbindung und ein einfaches REST (Representational State Transfer) basiertes Protokoll.

Ziel: Wir können eine erste Datenauswertung machen: Wie entwickelt sich die Luftfeuchtigkeit? Von was ist diese abhängig (z.B. Temperatur, Sonnenstand)?

Mehr: Welche Sensoren können das Forschungsschiff ergänzen? (Tabelle am Ende des Kapitels), wo könnte das Forschungsschiff ausgesetzt werden? (Tipp: es muss nicht im Wasser unterwegs sein: Auto, Tasche, Fahrrad, ...).

- Im Projekt „**Forschungsschiff informiert sich**“ holen wir Wetterinformationen für unseren Standort aus dem Internet.

Ziel: Technische Systeme kommunizieren untereinander.

Mehr: Welche weiteren nützlichen Informationen werden angeboten (Zugverspätungen, Staumeldungen, Aktienkurse). Welche Alternativen zum WLAN gibt es (UMTS, LoRa)?

12 Das Internet der Dinge anfassbar machen

- Im Projekt „**Forschungsschiff greift ein**“ schauen wir uns eine komplexere regelungstechnische Anwendung an. Wir nutzen Regeln (Wenn Das Dann Tue Dies) und binden ohne Programmierung unser Smartphone in die Anwendung ein.

Ziel: Nun geht es um den Mehrwert der Idee, wie kann ich diese im Alltag nutzen, wie kann ich diese an meine Bedürfnisse anpassen (was soll passieren wenn die Luftfeuchtigkeit im Keller zu hoch ist?)

Mehr: Wie kann ich weitere Dienste des Internets einbinden? (Twitter: wenn das Eichhörnchen eine Nuss aus der Futterstation entnommen hat).

- Im letzten Projekt „**Wearables**“ verwandeln wir die Flaschenpost in ein tragbares Kleidungsstück.

Ziel: Mobile Verwendung.

Ihr habt sicher ganz viele tolle Ideen ... ran an die Arbeit!

Die folgende Tabelle gibt einen zusammenfassenden Überblick über unsere Blaupausen und zeigt die dazu verwendeten Sensoren und Aktoren. Erweiterungen sind dank Grove¹²-System problemlos möglich. Zusätzliche Sensoren finden sich in Tabellen 3 bis 5 am Ende des Kapitels.

¹² exp-tech.de/pdf/products/grove/einstieg-grove.pdf

Das Internet der Dinge anfassbar machen

Tab. 2: Welche Interfacekomponenten werden genutzt?

Blaupause	Internet	LED	Dreh/ Drück	BME 280	LED- Matrix	Grove Relais (Digital)	Grove Sensor (Analog)
11.2.1 Hallo IoT-Kit		X	X				
11.2.2 Flaschenpost - Hallo Server	Server				X		
11.2.3 Forschungsschiff Hallo IoT	Client Thingspeak	Neo		X	(X)		
11.2.4. Forschungsschiff Hallo M2M	Client M2M-API yahoo	Neo			(X)		
11.2.5 Forschungsschiff Hallo Control	Client IFTTT	Neo				Pumpe	Feuchte
11.2.6. Wearables		Neo			(X)		Loudness

1.2.1 Hallo Arduino - Hallo IoT-Kit

Zur Erstellung unseres ersten Programmes starten wir die Arduino-Oberfläche und schauen uns kurz die einzelnen Komponenten an:

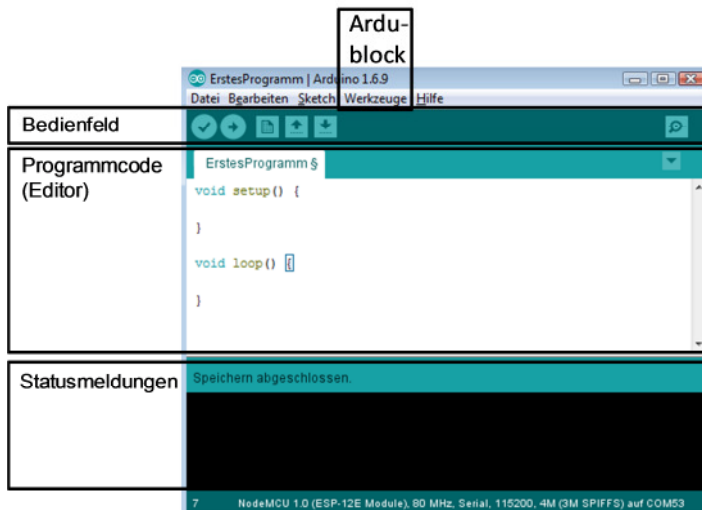






Abb. 4: Elemente der Arduino - IDE

Der Editor dient der Programmeingabe in C/C++ Hochsprache und unterstützt den Programmierer durch verschiedene farbliche Hervorhebung von Befehlen und Kommentaren.

Im Bedienfeld besteht die Möglichkeit, das Programm zu speichern , in Maschinocode zu übersetzen (überprüfen)  und schließlich auf die Zielhardware zu übertragen (hochladen, upload) . Um zur Laufzeit des Programmes interaktive Nutzereingriffe zu ermöglichen, besitzt die Entwicklungsumgebung ein eingebautes Terminal-Programm (hier als Serial Monitor bezeichnet, Ausgaben im Programm mit `Serial.print()`) .

Etwaige Syntax-Fehler, Warnungen oder Probleme beim Hochladen werden im Statusbereich angezeigt.

Anfänger bevorzugen sicher die grafische Programmierung mit Ardublock und rufen unter „Werkzeuge“->“Ardublock“ die entsprechende Oberfläche auf. Wir erstellen dort ein Ardublock-Programm, indem wir die entsprechenden Programmblöcke aus dem Menü auf der linken Seite auswählen und in den bereits vorhandenen Programmrumpf `main()` einfügen.

Das Internet der Dinge anfassbar machen

Die Einführung einer neuen Programmierumgebung erfolgt üblicherweise anhand eines einfachen Beispiels: Das Kultprogramm „Hallo-Welt“, welches einen kleinen Begrüßungstext ausgibt. Mangels eigenem Bildschirm beschränkt sich die Begrüßung bei unserem IoT-Kit auf ein einfaches Blinken der integrierten Leuchtdiode (LED). Diese Diode ist bei unserem Board mit dem Beinchen (Pin) GPIO0 (General Purpose I/O) des ESP-8266 verbunden.

Die `digitalWrite()` Funktion zur Ansteuerung der Leuchtdiode finden wir auf der linken Seite im Menü „Anzeigen/Aktoren“, die Warteanweisungen im Menü „Kontrollstrukturen“.

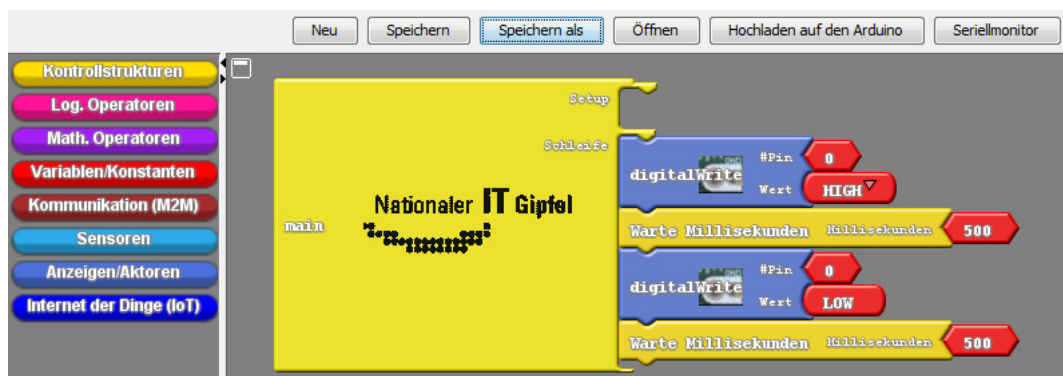


Abb. 5: Ardublock Programm Blink.abp

Nach Fertigstellung wird das erzeugte Programm mittels „Hochladen auf den Arduino“ auf das IoT-Kit geladen:

Damit wird der von Ardublock generierte C-Code in den Editor des Arduino kopiert und das Arduino-Programm sofort vom C-Compiler in Maschinencode übersetzt. Dieser Maschinencode wird anschließend über die USB-Verbindung auf das IoT-Kit geladen. Nach kurzem Uploadprozess befindet sich das Programm im Hauptspeicher des IoT-Kits und wird wie von Geisterhand sofort ausgeführt – die LED blinkt.

Hurra, wir haben unser IoT-Kit erfolgreich programmiert!

16 Das Internet der Dinge anfassbar machen

Zur Verdeutlichung des Programmaufbaus schauen wir uns das von Ardublock generierte Arduino-Programm etwas genauer an:

```
1: void setup(){ // Initialisierung, wird nur einmalig ausgeführt
2:   pinMode(0,OUTPUT); // GPIO0 ist Ausgang
3: }
4:
5: void loop() { // Kontinuierliche Wiederholung
6:   digitalWrite(0,HIGH); // LED an
7:   delay(500); // 500 ms warten
8:   digitalWrite(0,LOW); // LED aus
9:   delay(500); // 500 ms warten
10: }
```

Ein typisches Arduino-Programm (auch Sketch genannt) besteht aus zwei Funktionen. Im Unterprogramm `setup()` (Zeilen 1-3) werden alle Initialisierungsaufgaben einmalig ausgeführt. Alle im Unterprogramm `loop()` (Zeilen 5-10) stehenden Befehle werden dagegen zyklisch wiederholt. Der Programmblock eines Unterprogrammes wird immer mit geschweiften Klammern `{}` umschlossen. Kommentare werden mit `//` eingeleitet. Die in der ersten Spalte angegebenen Programmzeilen dienen nur der vereinfachten Referenzierung im Beschreibungstext und sind nicht Bestandteil des eigentlichen Programmcodes.


Da eine Spannungsausgabe auf einem Pin natürlich die Umgebung (hier in Form der angeschlossenen LED) aktiv beeinflusst und ggf. Schaden anrichten kann, muss dieser Pin im `setup()` erst explizit als Ausgang frei geschaltet werden (Zeile 3 des Programmes).

In unserem Beispiel erfolgt die abwechselnde Ausgabe eines hohen Spannungspegels (HIGH, 3.3 V, Zeile 6) und eines niedrigen Spannungspegels (LOW, 0V, Zeile 8) an dem Pin mit der Nummer 0, d.h. dem Pin, an dem die LED angeschlossen ist.

Unser Mikrocontroller ist in der Lage, diese Befehle mehrere tausend Male in der Sekunde zu wiederholen. So schnell kann das Auge nicht folgen, weshalb unser Programm künstlich durch Wartezeiten (`delay`) verlangsamt wird (Zeilen 7 und 9).

Das Internet der Dinge anfassbar machen

Natürlich können wir statt der grafischen Ardublock-Programmierung auch gleich den C-Code in den Editor eingeben oder wir laden den C-Code über „Datei“ -> „Öffnen vom Datenträger“ in den Editor.

Zur erneuten Ausführung übertragen wir unser Programm in der Arduino-GUI mittels Hochladen  an die Zielhardware.

Ardublock ist dann überflüssig - vorausgesetzt, wir sind mit der Programmierung in C/C++ vertraut. Auch können wir mit Ardublock einen Rumpf erzeugen und diesen im Editor erweitern. So bietet unser Arduino-System größtmögliche Flexibilität sowohl für Anfänger, als auch für Fortgeschrittene.

Alle vorgestellten Listings finden sich auch online unter <http://div-konferenz.de/events/hackathon>.

Übrigens: Ein heruntergeladenes Programm ist auch nach Ausschalten und Wiederinbetriebnahme des Kits vorhanden. Erst ein erneutes Hochladen ändert die Programmausführung.

Die Komponenten des IoT-Kits

Das Kit besitzt neben der bereits bekannten LED an Pin 0 einen Dreh/Drückknopf (bei Ardublock zu finden im Menü „Sensoren“) und zwei RGB-NeoPixel (farbige Leuchtdioden, im Menü „Aktoren“) sowie verschiedene Umweltsensoren.

Die **NeoPixel** lassen sich in Farbe und Intensität verändern. Jede Farbe wird durch die Anteile der Grundfarben rot / grün / blau beschrieben. Unsere Anzeigefunktion besitzt also drei Parameter für die Leuchtkraft der jeweiligen Grundfarbe (0...63). Die maximale Helligkeit ist im Ardublock auf einen Farbwert von 63 beschränkt, da ein NeoPixel sonst unserer Taschenlampe Konkurrenz machen kann und eventuell die Augen blendet.

18 Das Internet der Dinge anfassbar machen

Der folgende Algorithmus erkennt das Drücken des Bedienknopfes und lässt die rechte LED weiss aufleuchten. Ist der Knopf nicht gedrückt, so wird die LED ausgeschaltet.

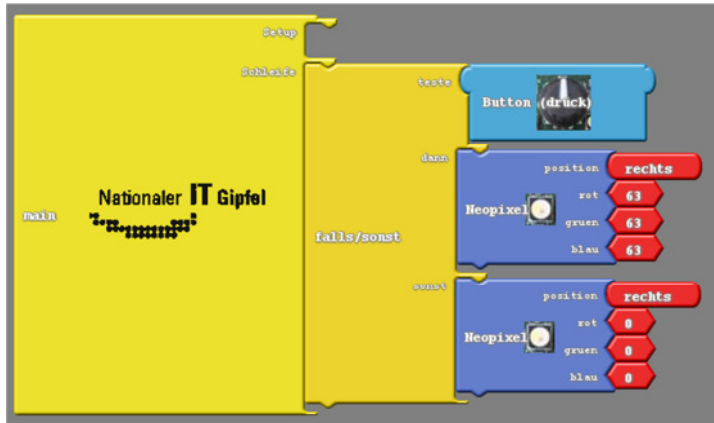


Abb. 6: Ardublock-Programm ButtonNeopixel.abp

Auch hier wieder ein kurzer Blick hinter die Kulissen des generierten C-Programmes:

```
1: #include <Adafruit_NeoPixel.h> // Bibliothek einbinden
2: Adafruit_NeoPixel
3:     pixels=Adafruit_NeoPixel(2,13, NEO_GRBW+NEO_KHZ800);
4:
5: void setup() { // Initialisierung, wird nur einmalig ausgeführt
6:     pixels.begin(); // Neopixel initialisieren
7:     delay(100); // kurz warten
8: }
9:
10: void loop() { // Kontinuierliche Wiederholung
11:     if (digitalRead(2)==LOW) // Button schaltet gegen Masse
12:     { // Knopf gedrückt -> Neopixel rechts -> RGB an -> weiss
13:         pixels.setPixelColor(0,63&0x3F,63&0x3F,63&0x3F);
14:         pixels.show();// Farbe ausgeben
15:     } else { // Knopf nicht gedrückt -> Neopixel aus
16:         pixels.setPixelColor(0,0&0x3F,0&0x3F,0&0x3F);
17:         pixels.show();//Farbe ausgeben
18:     }
19: }
```

Zur Ansteuerung der NeoPixel wird ein spezielles Protokoll verwendet, dieses wird über eine Bibliothek bereitgestellt (include Anweisung Zeile 1).

Der **Dreh/Drückknopf** ist mit drei digitalen Eingangspins verbunden. Ein Pin (GPIO2) liegt immer am High-Pegel, es sei denn, wir drücken auf den Knopf. Dann schaltet dieser Pin gegen Masse (Low). Abfragen lässt sich der aktuelle Signalzustand einfach über einen `digitalRead()`-Befehl (Zeile 11).

Das Internet der Dinge anfassbar machen

Komplizierter ist die Sache mit der Drehfunktion, aber Dank des bereitgestellten Ardublock-Moduls lässt sich diese letztendlich genauso einfach bedienen, wie die Drückfunktion. Für Details sei an die C-Code Version der Blaupausen verwiesen.

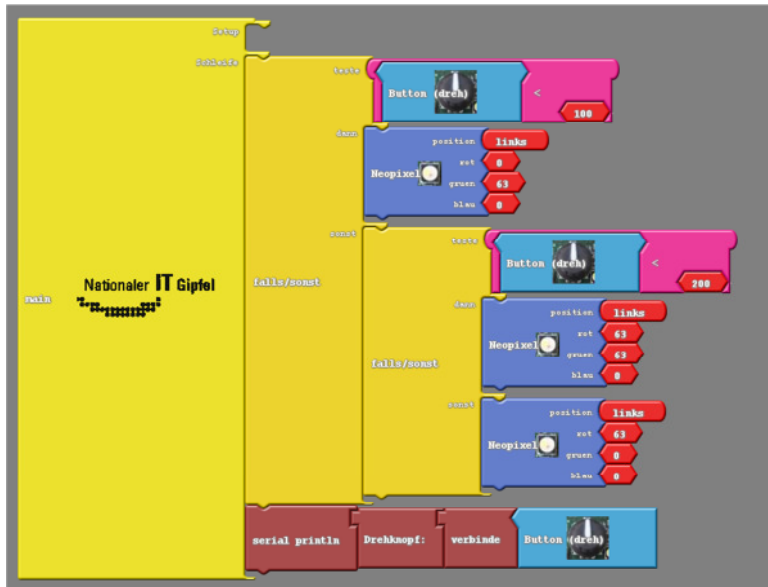


Abb. 7: Ardublock ButtonDreh.abp

Das obige Programm liest die Stellung des Drehkopfes ein. In einer geschachtelten Fallunterscheidung wird das linke NeoPixel in Abhängigkeit der Drehposition auf eine andere Farbe gesetzt. Der Drehknopf beginnt bei Programmstart mit Position Null. Eine Rechtsdrehung erhöht, eine Linksdrehung dekrementiert die vom Sensor zurückgelieferte Positionsangabe. Ist der Wert kleiner 100, so leuchtet unser NeoPixel grün, bei einem Wert zwischen 100 und kleiner 200 gelb, darüber rot. Die letzte Zeile gibt uns zur Kontrolle die Positionsangabe auf der seriellen Schnittstelle aus. Diese Ausgabe lässt sich über den Menüpunkt „Seriellmonitor“ aktivieren.

Hurra, die Ein-/Ausgabemöglichkeiten sind klar!

Übrigens: Mit dem Rotary-Encoder lassen sich z.B. die Drehgeschwindigkeiten von Motoren ermitteln, oder die Position eines drehbaren Roboterarmes bestimmen.

1.2.2 Die digitale Flaschenpost – Hallo Server

Natürlich lassen sich auch mehrere LEDs zu einer Matrix zusammenschalten. Bei unserem Kit geschieht dies durch Nutzung eines externen Erweiterungsboards: Das CharlieWing Matrix-Display¹³. Dieses wird auf die FeatherWing Steckleisten unseres Kits gesetzt und dient der erweiterten Visualisierung.

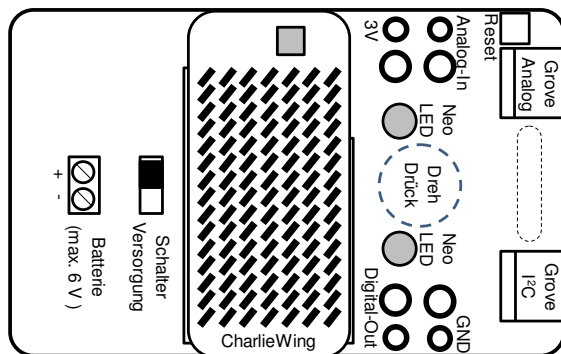


Abb. 8: IoT-Kit mit LED-Matrix

Durch Verwendung der entsprechenden Bibliotheksfunktionen gelingt so auch die Textausgabe des „Hallo IoT-Kit“ mit wenigen Programmzeilen:

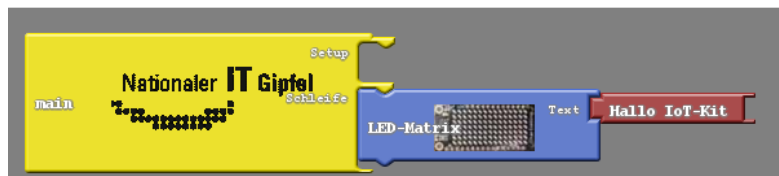


Abb. 9: Ardublock HalloMatrix.abp

Allerdings ist die Ausgabe eines festen Textes nach einiger Zeit recht langweilig und wir wünschen uns mehr Flexibilität:

Da wir über ein internetfähiges Kit verfügen wäre es doch toll, wenn wir den auszugebenden Text per Web-Interface vom Handy aus ändern könnten. Dazu werden wir mit wenigen Programmzeilen einen eigenen HTTP – Server implementieren und per Internet-Browser darauf zugreifen.

¹³ learn.adafruit.com/adafruit-15x7-7x15-charlieplex-led-matrix-charliewing-featherwing

Das Internet der Dinge anfassbar machen

Auch hier erleichtert ein entsprechendes Ardublock-Modul die Umsetzung des Vorhabens:

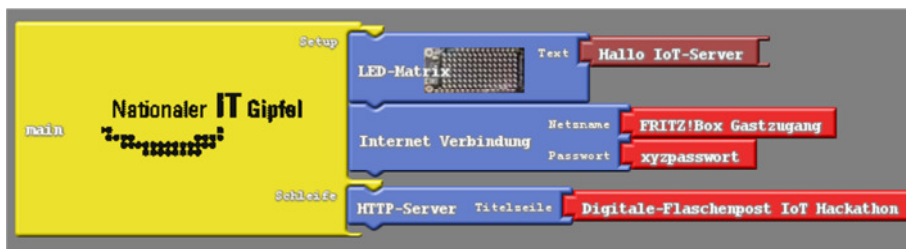


Abb. 10: Ardublock HaloMatrixServer.abp

Das Programm begrüßt uns mit einer Startmeldung „Hallo-IoT-Server“ und versucht dann, den Zugang zum WLAN-AccessPoint herzustellen. Ist dies gelungen, erscheint die vom Accesspoint zugewiesene IP-Adresse in der Anzeige. Melden wir uns mit unserem Handy, oder einem anderen Rechner, am selben WLAN-AccessPoint an, so können wir die Homepage unseres Moduls unter dieser IP-Adresse erreichen.

Eine Homepage besteht normalerweise aus einer Textdatei in einem speziellen Format (Hypertext Markup Language, HTML). Da unser System keine Dateien verwalten kann, erstellt Ardublock stattdessen ein Textfeld mit dem gewünschten Inhalt. Die angegebene Titelzeile erscheint als Überschrift auf der Seite:

```
1: const char INDEX_HTML[] =
2:   "<html>"
3:   "<head>"
4:     "<title>ESP8266 Web Form Demo</title>"
5:   "</head>"
6:   "<body>"
7:     "<h1>Digitale - Flaschenpost IoT Hackathon </h1>"
8:     "<FORM action=\"/\" method=\"post\">"
9:       "<P>"
10:        "Deine Nachricht<br>"
11:        "<INPUT type=\"text\" name=\"message\"><br>"
12:        "<INPUT type=\"submit\" value=\"Send\">"
13:      "</P>"
14:    "</FORM>"
15:  "</body>"
16: "</html>";
```

22 Das Internet der Dinge anfassbar machen

Daneben enthält unsere HTML-Seite mit dem Input-Tag ein im Browser des Clients editierbares Textfeld. Der Inhalt dieses Feldes wird über den Send-Button an unseren Web-Server übermittelt. Über diesen Mechanismus aktualisieren wir unseren anzuzeigenden Text.

Stolz präsentieren wir die erste eigene Homepage:

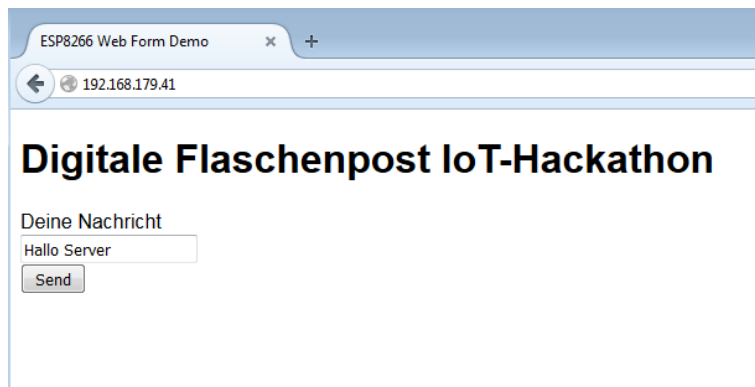


Abb. 11: Homepage unserer Flaschenpost

In einer Flasche verpackt, kann das Kit so eine digitale Nachricht an den Empfänger senden – ähnlich wie eine Flaschenpost in unserer analogen Welt.

Hurra, wir haben den ersten eigenen Web-Server erstellt!

Übrigens: Es gibt keine Sicherheitsfunktionen. Wer unsere IP-Adresse kennt, der kann uns so eine Nachricht schicken!
(Tipp: Wir können die Flaschenpost z.B. mit einem geheimen Code vor dem eigentlichen Text schützen!)

1.2.3 Unser Forschungsschiff – Hallo IoT

Im folgenden Abschnitt wollen wir unsere Flaschenpost zusätzlich über verschiedene Sensoren direkt mit der Außenwelt verbinden (Datenlogger) und als Forschungsschiff auf Reisen schicken. Einfaches Beispiel dafür ist der eingebaute Bosch-Sensor BME280¹⁴. Dieser ist in der Lage, die Temperatur, die Luftfeuchte und den Luftdruck der Umwelt zu bestimmen. Über eine digitale Schnittstelle, den sogenannten I²C-Bus (Inter IC-Bus)¹⁵, lassen sich die Messwerte vom Mikroprozessor auslesen und anzeigen. Ein spezielles Schnittstellenmodul „verbinde“ wandelt dabei den numerischen Messwert in einen Text, der über die Matrix ausgegeben wird. Ist keine Matrix vorhanden, kann stattdessen auch der serial println-Befehl zur Ausgabe auf die serielle Schnittstelle genutzt werden.

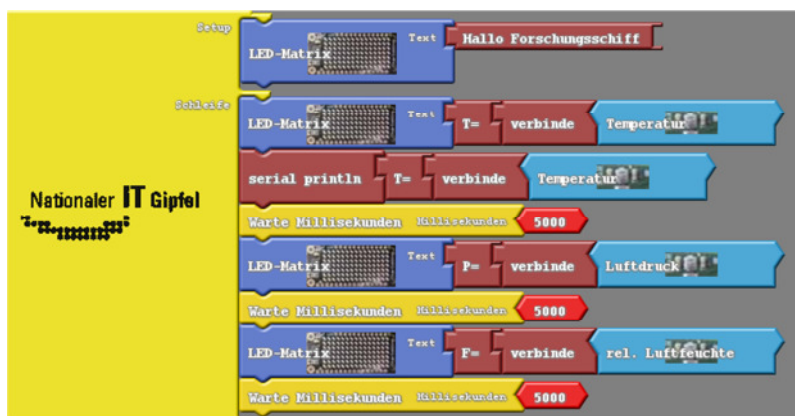


Abb. 12: ForschungsschiffMatrix.abp

¹⁴ https://www.bosch-sensortec.com/bst/products/all_products/bmp280

¹⁵ <http://www.elektronik-magazin.de/page/der-i2c-bus-was-ist-das-21>

24 Das Internet der Dinge anfassbar machen

Zu einem Datenlogger gehört natürlich neben der eigentlichen Messdatenerfassung noch mehr, nämlich die Speicherung der gemessenen Werte. Üblicherweise geschieht dies auf einem USB-Stick oder einer SD-Karte. Da wir ein internetfähiges Kit besitzen, können wir glücklicherweise auf diese externen Speicher verzichten und unsere Daten direkt dem Internet anvertrauen. In diesem Fall spricht der Informatiker von einer Cloud-Lösung, bei der ein im Internet verfügbarer Server unsere Daten entgegennimmt und für uns archiviert. Hierzu existieren eine Reihe von Diensten verschiedener Anbieter¹⁶.

Im Rahmen des Hackathon verwenden wir die Open-Source Plattform Thingspeak. Zur Nutzung dieses Dienstes müssen wir uns vorher beim Server registrieren. Der offizielle Server ist weltweit unter thingspeak.com erreichbar. Alternativ dazu gibt es eine am Umwelt-Campus gehostete lokale Installation unter der URL www.thingspeak.umwelt-campus.de. Diese ist nur im Campus-Netz erreichbar, alle unsere Daten bleiben lokal.

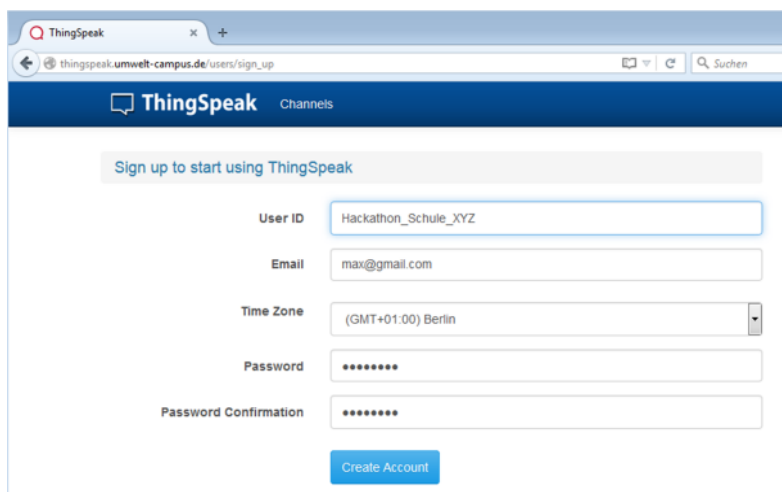
The image shows a web browser window displaying the registration page for Thingspeak. The browser's address bar shows the URL 'thingspeak.umwelt-campus.de/users/sign_up'. The page has a dark blue header with the 'Thingspeak' logo and the word 'Channels'. Below the header, there is a light blue box with the text 'Sign up to start using ThingSpeak'. The registration form consists of several fields: 'User ID' with the value 'Hackathon_Schule_XYZ', 'Email' with 'max@gmail.com', 'Time Zone' with a dropdown menu set to '(GMT+01:00) Berlin', 'Password' and 'Password Confirmation' both filled with seven asterisks. At the bottom of the form is a blue button labeled 'Create Account'.

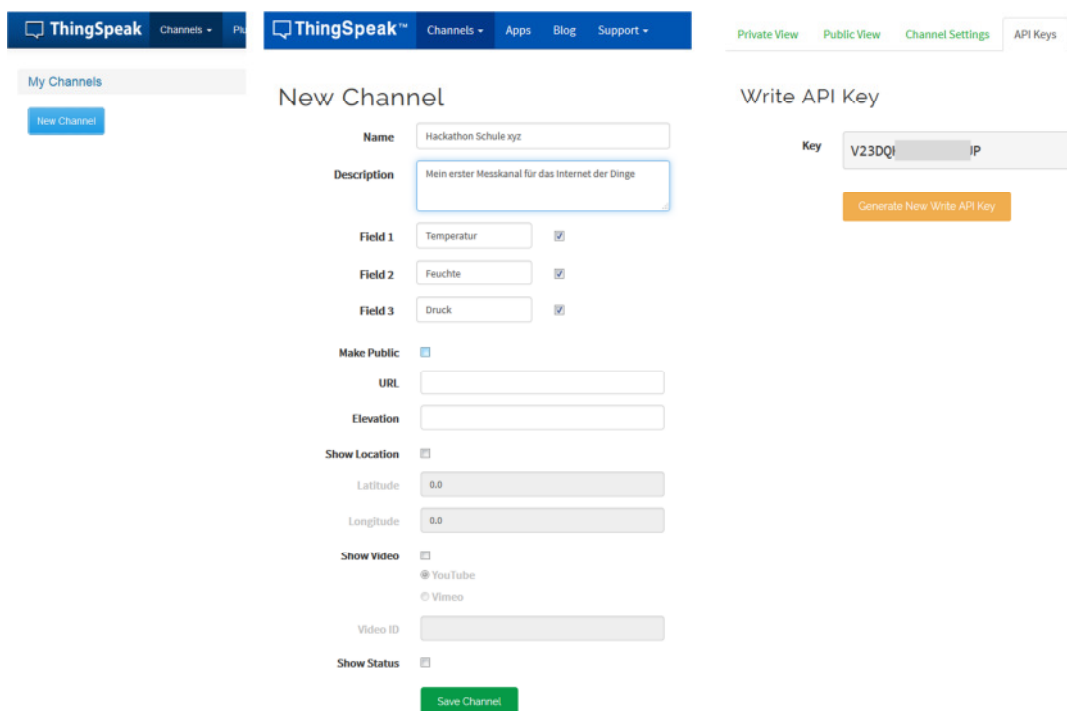
Abb. 13: Registrierung Thingspeak-Cloud

¹⁶ phant.io, thingspeak.com

Das Internet der Dinge anfassbar machen

Anschließend können wir unter „Channels“ -> „New Channel“ den ersten Messkanal für den Hackathon konfigurieren. Dort erstellen wir drei Feldeinträge für Temperatur, Feuchte und Luftdruck. Wichtig ist die Reihenfolge, denn ein Feld wird bei der späteren Datenspeicherung durch den Feldindex adressiert. Wer möchte, kann die Messdaten später auch für andere Nutzer sichtbar machen (Make Public-Flag).

Nach Speichern der Konfiguration steht der Kanal ab sofort zum Zugriff bereit.



The screenshot shows the ThingSpeak web interface. On the left, there's a 'My Channels' sidebar with a 'New Channel' button. The main area is titled 'New Channel' and contains a form with the following fields: 'Name' (Hackathon Schule xyz), 'Description' (Mein erster Messkanal für das Internet der Dinge), 'Field 1' (Temperatur), 'Field 2' (Feuchte), 'Field 3' (Druck), 'Make Public' (checkbox), 'URL', 'Elevation', 'Show Location' (checkbox), 'Latitude' (0.0), 'Longitude' (0.0), 'Show Video' (checkbox), 'Video ID', and 'Show Status' (checkbox). A 'Save Channel' button is at the bottom. On the right, there's a 'Write API Key' section with a 'Key' field containing 'V23DQ!' and 'JP', and a 'Generate New Write API Key' button.

Abb. 14: Erstellung eines Messkanals

Bevor wir aber Daten dort ablegen können, benötigen wir eine Zugangsberechtigung, den sogenannten **API-Key**. Nur wer in Besitz dieses Schlüssels ist, kann Daten schreiben. Sicherheit spielt beim IoT eine große Rolle. Wir notieren uns also unseren Key und setzen ihn später an geeigneter Stelle im Programm ein (oder kopieren ihn mit cut & paste aus der Browseranzeige).

Nach erfolgreicher Konfiguration können die Daten mittels einfacher HTTP- GET Nachricht in der Datenbank des Servers abgelegt werden. Zum Test lässt sich diese GET-Nachricht von jedem Webbrowser aus generieren. Dazu im Browser die URL

`http://api.thingspeak.com/update?api_key=V23DQyyyyUP&field1=34.5`

aufrufen, wobei der API-Key an den eigenen Kanal angepasst werden muss. Der Server (host) übernimmt die Daten für die Temperatur (hier 34.5 für das erste Feld) und antwortet mit der laufenden Nummer des aktuellen Eintrags in seiner Datenbank.

Eine solche GET Nachricht lässt sich per entsprechendem Ardublock auch einfach durch unser IoT-Kit absenden.

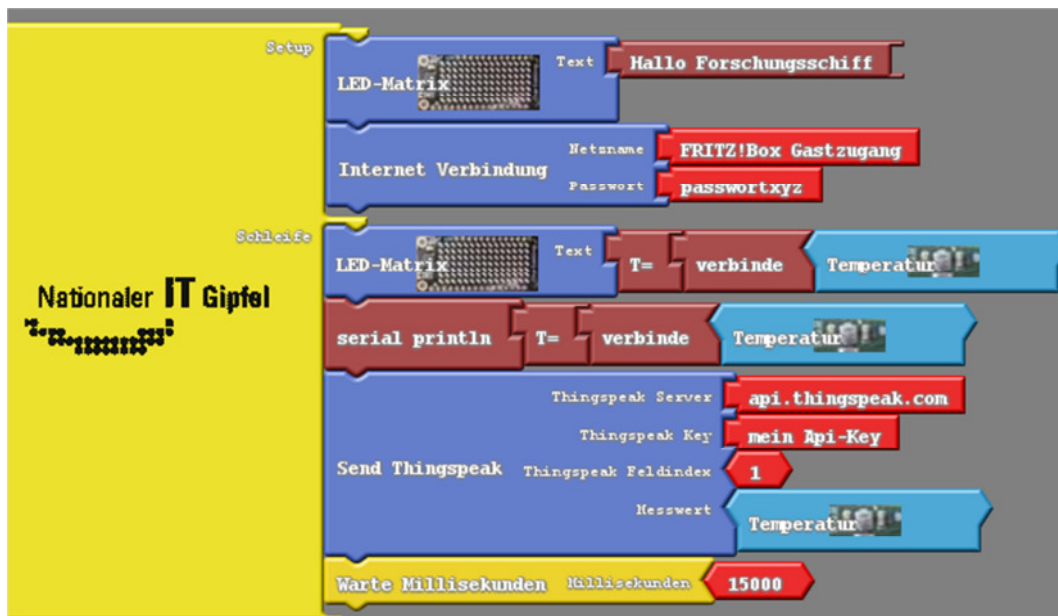


Abb. 15: ForschungsschiffThingspeak.abp

Zur individuellen Festlegung des Messkanals muss natürlich der API-Key im Thingspeakblock an unseren persönlichen Key angepasst werden. Außerdem geben wir den zu aktualisierenden Feldindex (1 für den ersten Eintrag, Temperatur) an.

Der Server Thingspeak.com lässt nur Aktualisierungen von mehr als 15 Sekunden zu. Deshalb fügen wir noch einen Wartebefehl ein.

Das Internet der Dinge anfassbar machen

Kontrollieren können wir den Erfolg auf der Thingspeak-Seite im Internet (oder im Falle von thingspeak.umwelt-campus.de nur im Intranet):

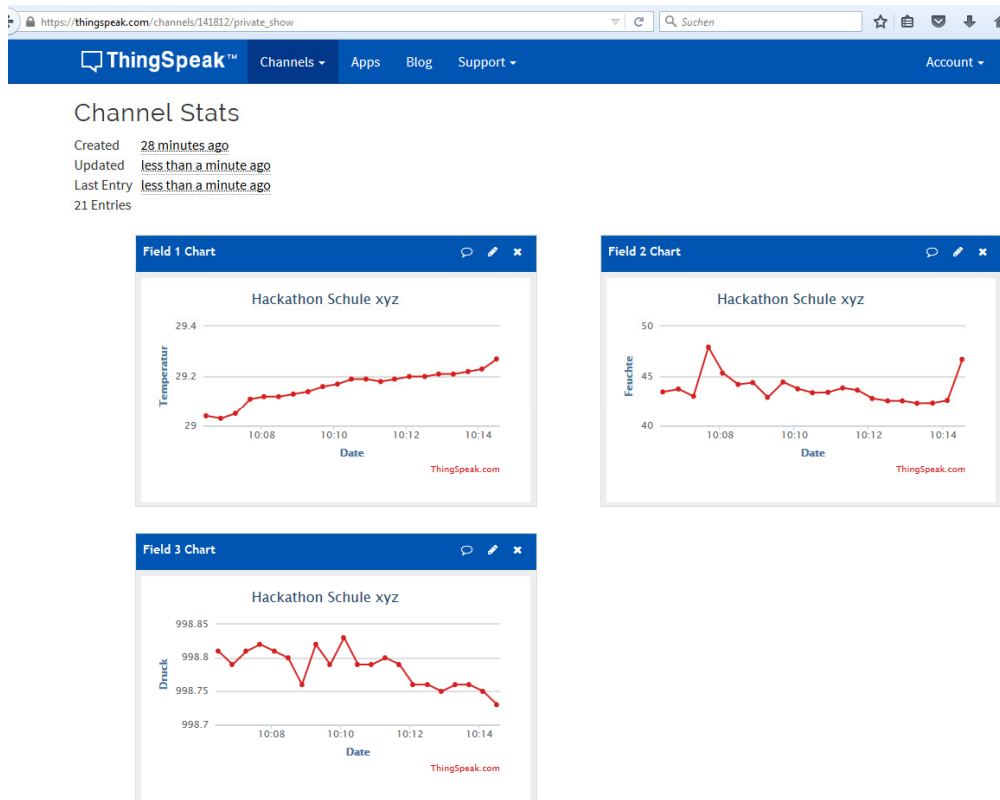


Abb. 16: Visualisierung eines Messkanals

Dort sind unsere Messwerte ab sofort weltweit verfügbar. Damit haben wir mit wenig Programmcode einen internetfähigen Datenlogger für Umweltinformationen erstellt.

Hurra, wir haben unsere Flasche in ein Forschungsschiff mit IoT-Anbindung verwandelt!

Übrigens: Die Sensorik lässt sich ganz einfach über den Grove-Connector erweitern. Denkbar sind alle in einer Tabelle am Ende des Kapitels aufgeführten Sensoren.

Wir sollten auch den Energiebedarf im Auge haben (mehr dazu unter [Maker & Hacker](#) am Ende des Kapitels).

1.2.4 Forschungsschiff informiert sich – Hallo M2M

Natürlich kann ein IoT-Device auch selbst Informationen aus dem Internet abfragen. Dazu definieren die Dienstbringer spezifische Schnittstellen (API, Application Programming Interface), über die die Informationen bereitgestellt werden können. Das Prinzip ist ähnlich, wie beim Aufruf normaler Webseiten, nur die Antwort erfolgt in einem maschinenlesbaren Format (z.B. XML, Extensible Markup Language). Als Beispiel verwenden wir die Wettervorhersage von yahoo¹⁷.

Mit Hilfe des bereits bekannten HTTP-GET können die aktuellen Wetterbedingungen für beliebige geografische Positionen abgerufen werden. Der Ort wird dabei über eine Identifikationsnummer (woeid) festgelegt¹⁸. Das folgende Programm ermittelt die aktuelle Windgeschwindigkeit für Saarbücken (woeid 690631, ggf. anpassen):

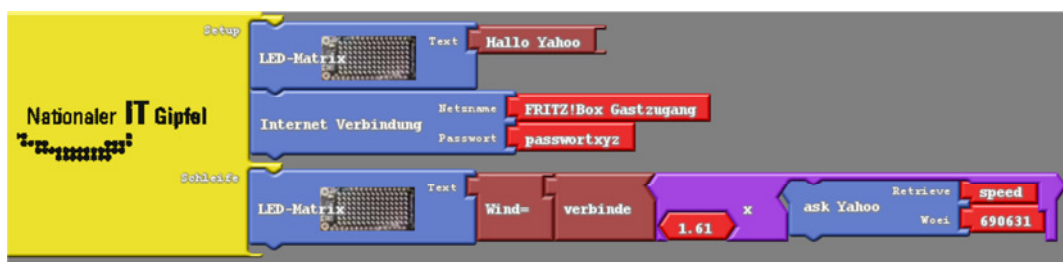


Abb. 17: ForschungsschiffYahoo.abp

Yahoo liefert die Daten in Meilen pro Stunde, unsere Ausgabe erfolgt nach Umrechnung in km/h.

Hurra, wir können die Wettervorhersage nutzen!

¹⁷ <https://developer.yahoo.com/weather/>

¹⁸ <http://woeid.rosselliot.co.nz/>

Das Internet der Dinge anfassbar machen

Übrigens: Wenn wir die Windgeschwindigkeit mit einer Fallunterscheidung (if ... then ... else) überprüfen, können wir ab einer bestimmten Windgeschwindigkeit einen Alarm auslösen (z.B. durch Setzen eines digitalen Ausgangs ähnlich wie bei der blinkenden LED). Damit lässt sich eine einfache Überwachung z.B. einer Markise realisieren.

1.2.5 Forschungsschiff greift ein - Hallo Control

Die bedingte Ausführung von Programmteilen lässt sich prima dazu nutzen, um per Rückkopplung in ein System einzugreifen. Der Ingenieur spricht dann von einer Regelung. Als einfaches Beispiel bauen wir uns gleich eine automatische Bewässerung für die Blumentöpfe in der Fensterbank. Dazu benötigen wir einen Sensor, welcher die Leitfähigkeit der Erde ermittelt und als Maß für die Feuchtigkeit genutzt werden kann. Ist die Erde zu trocken, steuern wir eine Pumpe an, welche Wasser in den Topf befördert.

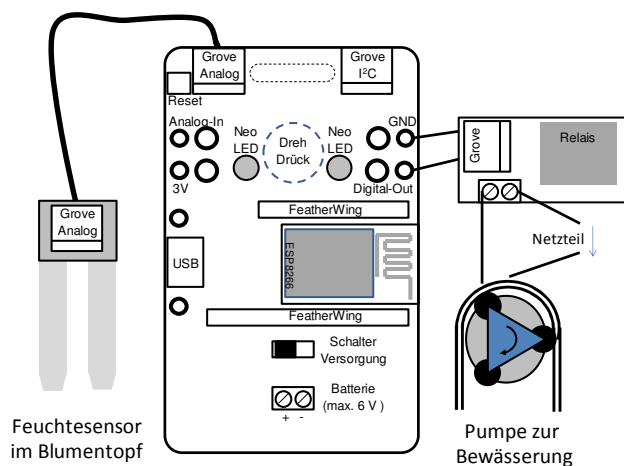


Abb. 18: Das Forschungsschiff als Bewässerungsregler

Mit unserer bisher gesammelten Erfahrung ist das programmtechnisch einfach umzusetzen:

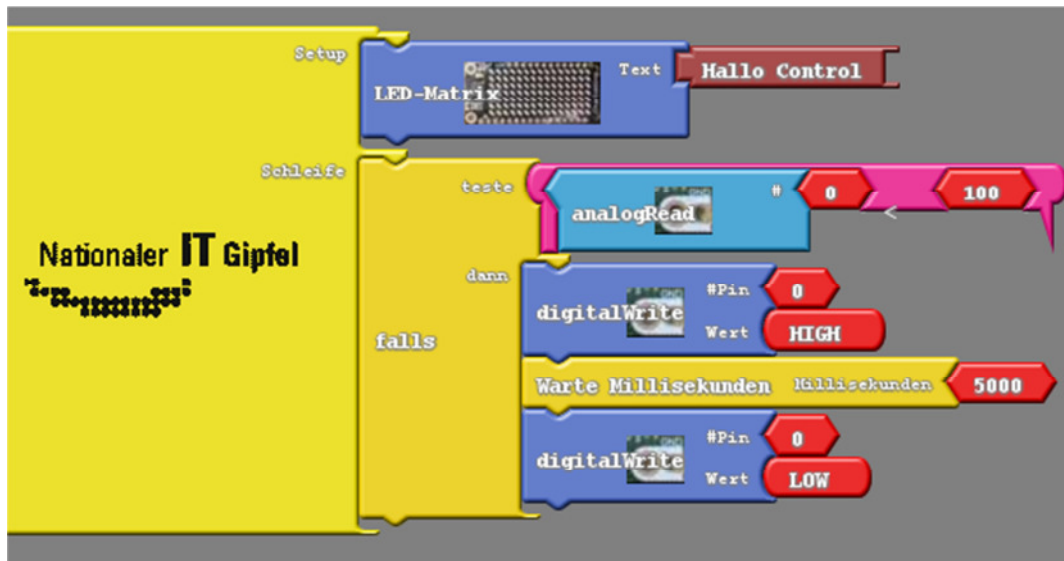


Abb. 19 ForschungsschiffControl.abp

Leider kann der Mikrocontroller nur kleine Lasten wie z.B. eine LED schalten. Hier setzen wir deshalb ein Relais ein. In unserem Fall haben wir das Relais zur Pumpenansteuerung mit dem bereits bekannten I/O-Pin GPIO0 verbunden (Bananensteckerbuchsen I/O auf der rechten Seite). Das Relais wiederum schaltet die Verbindung zwischen Pumpe und Netzteil. **Achtung: wir verwenden natürlich nur Kleinspannungen – niemals die 230V aus der Steckdose schalten!**

Innerhalb der Regelschleife lesen wir das vom Feuchtesensor¹⁹ gelieferte Spannungssignal. Dieses wird vom Analog-Digital-Wandler (ADC) des ESP in einen Wert zwischen 0 und 1023 gewandelt. Ist der Wert kleiner als eine Grenze (z.B. 100 für Trocken), so wird die Pumpe für kurze Zeit angeschaltet. Die Erde wird feucht, die Leitfähigkeit steigt, der Sensor liefert hoffentlich höhere Werte für die Leitfähigkeit.

¹⁹ http://seeedstudio.com/wiki/Grove_-_Moisture_Sensor

Das Internet der Dinge anfassbar machen

Kritisch wird die Anwendung dann, wenn unser Vorratsgefäß leer ist. Dann trocknet die Erde weiter aus und wir erreichen eine untere Alarmgrenze. In diesem Fall wollen wir wieder unsere IoT-Fähigkeiten ins Spiel bringen und eine Meldung über Twitter, E-Mail oder SMS generieren.

Dazu bietet sich der Internet-Dienst „If this then that“ (IFTTT)²⁰ an.

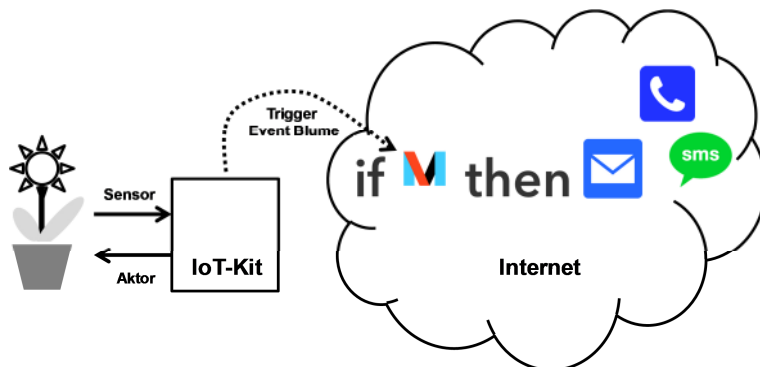


Abb. 20: Informationsfluss IFTTT

Auch hier müssen wir uns natürlich vorher registrieren:

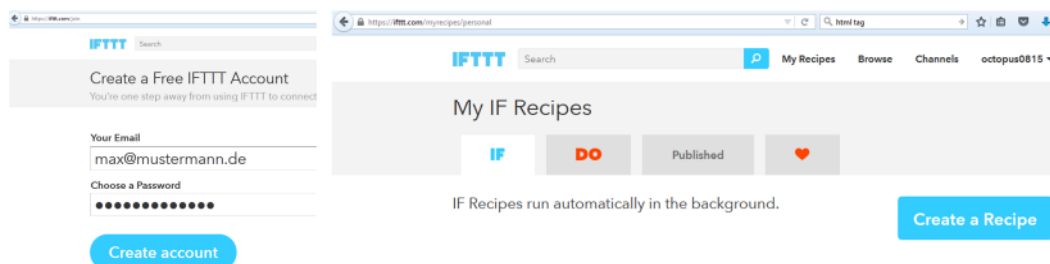


Abb. 21: Registrierung IFTTT

Mittels komfortabler Weboberfläche können wir anschließend mit einfachem Knopfdruck ein Rezept (Recipe) der Form IF then DO.... erstellen (d.h. konfigurieren, nicht programmieren). In unserem Fall ist die Triggerbedingung (IF) ein vom IoT-Kit erzeugter Event „Blume“ auf dem Makerkanal:

²⁰ <https://ifttt.com>

32 Das Internet der Dinge anfassbar machen

Create a Recipe

if **this** then **that**

Choose Trigger Channel step 1 of 7
Showing Channels that provide an Action Trigger [View all Channels](#)

Maker

Maker
Webhooks
Webhooks Maker

Complete Trigger Fields step 3 of 7
Receive a web request

Event Name

Blume

The name of the event, like "button_pressed" or "front_door_opened"

Create Trigger

Abb. 22: Ereignis (Event) definieren

den Zugriffskey finden wir anschließend unter <https://ifttt.com/maker> und als Action lassen wir uns eine E-Mail schicken:

if **this** then **that**

Choose Action Channel step 4 of 7 [back](#)
Showing Channels that provide at least one Action [View all Channels](#)

email

email
Event Digest
Email

Complete Action Fields step 6 of 7
Send me an email

Subject

The event named " `EventName` " occurred on the Maker Channel

Body

What: `EventName`

When: `OccurredAt`

Extra Data: `Value1` , `Value2` , `Value3` ,

Create Action

Abb. 23: Action Mailversand

Damit können wir uns jetzt im Urlaub beruhigt zurücklehnen. Die Blumen sind versorgt – und sollte mal was schiefgehen, erhalten wir eine mail und können die Nachbarn aktivieren.

Das Internet der Dinge anfassbar machen

Das Ardublock-Programm soll den Trigger nur einmalig auslösen, wir wollen ja nur eine Benachrichtigungsmail erhalten. Dies realisieren wir einfach mit einer digitalen Variable „AlarmScharf“, die wir im Setup scharf stellen und nach der ersten Trigger-Nachricht löschen.

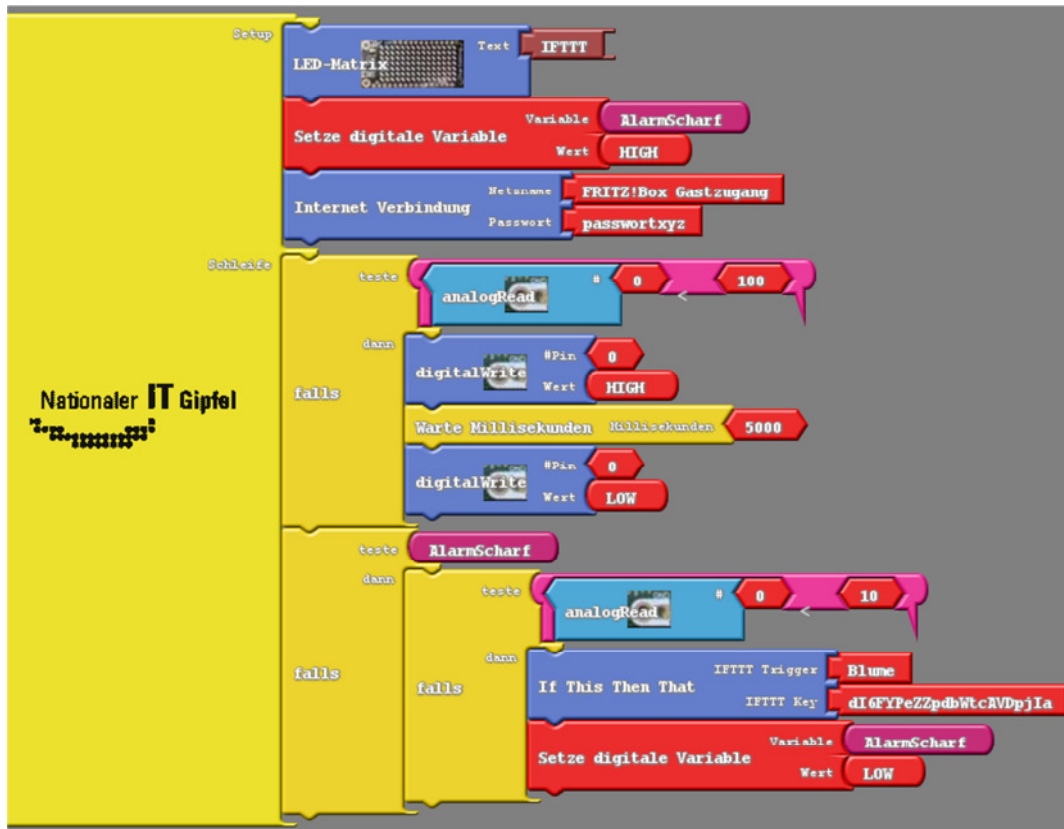


Abb. 24: ForschungsschiffIFTTT.abp

Hurra, wir haben eine Blumenbewässerung realisiert!

Übrigens: So ähnlich lassen sich viele Alltagsaufgaben überwachen. Zum Beispiel die Luftfeuchtigkeit im Keller durch Ansteuerung eines Lüfters, oder die Alarmanlage mit Überprüfung der Fensterkontakte und Ansteuerung einer Sirene.

1.2.6 Wearables – Forschungsschiff kleidet sich

Die einzeln ansteuerbaren bunten NeoPixel eröffnen viele neue Anwendungen. Als kleine Knöpfe lassen sie sich problemlos in Kleidungsstücke integrieren. Dazu gibt es leitfähiges Nähmaterial, mit denen die Anzeigeelemente geschickt befestigt und mit Energie versorgt werden können. Unser IoT-Kit ist aber an sich schon so kompakt, dass es problemlos am Körper befestigt werden kann. Mit einer mobilen Energieversorgung über Batterien können wir unser „Hallo Arduino“-Projekt sofort recyceln: Als Namensschild zur Kennzeichnung der Coaches am Hackathon. Fügen wir etwas Sensorik in Form eines Mikrofons hinzu, so haben wir eine mobile Lichtorgel. Je nach Umgebungslautstärke leuchten die Neopixel in einer anderen Farbe.

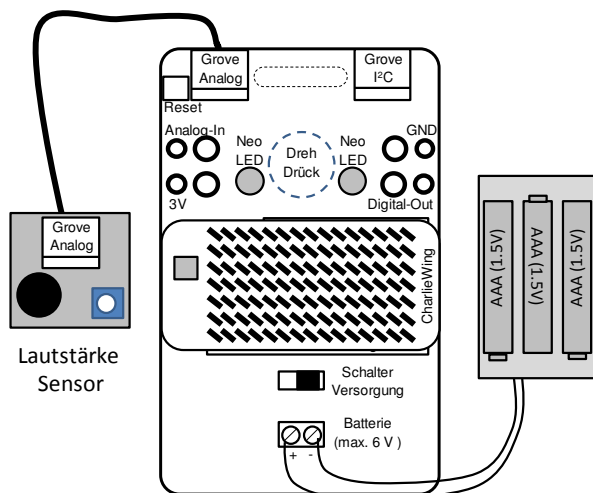


Abb. 25: Tragbares IoT-Kit mit Lärmsensor

Zur Nutzung der Batterie müssen wir den Wahlschalter zur Spannungsversorgung nach rechts stellen. Weitere Hinweise zum Batteriebetrieb finden sich in der C-Code Version der Blaupausen.

Übrigens: Wer es auffällig liebt, kann die CharlieWing-Matrix auch durch ein NeopixelWing ersetzen und diese statt des einzelnen Neopixels ansteuern (AmpelNeoWing.abp). So wird unsere Ampel unübersehbar.

Das Internet der Dinge anfassbar machen

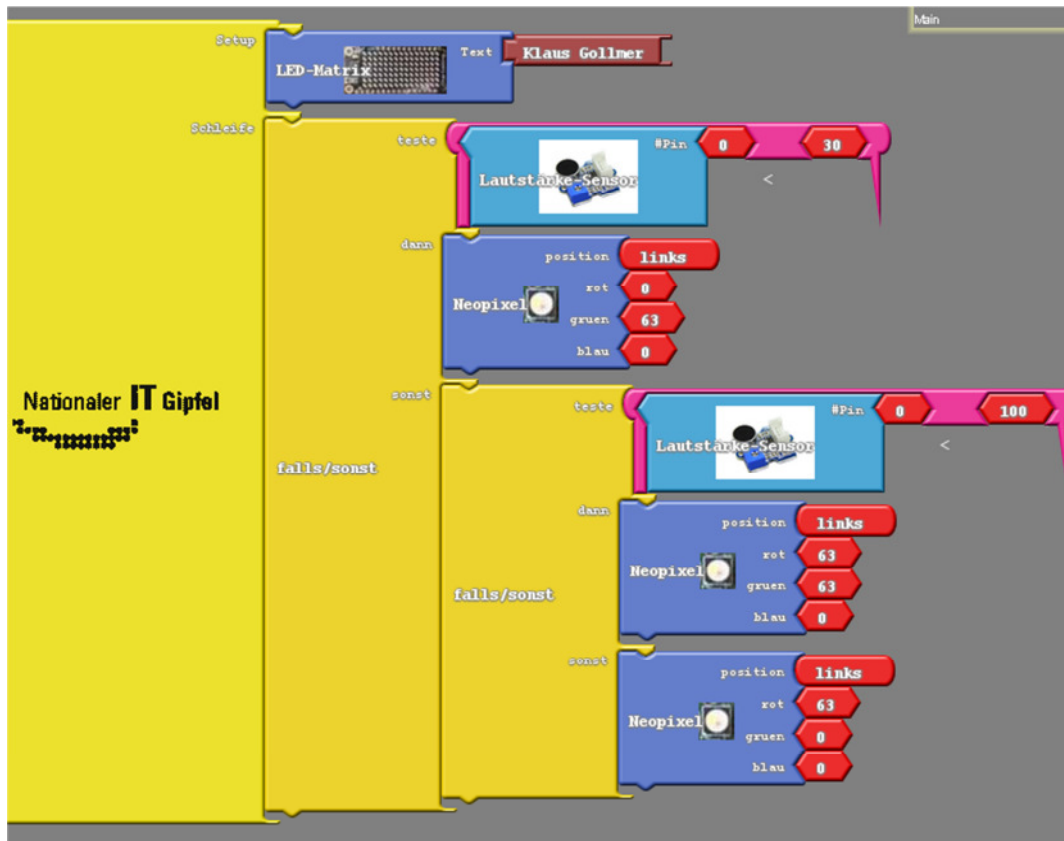


Abb. 26 AmpelNeoNamen.abp

Hurra, unser Kit ist wie ein Kleidungsstück tragbar!

Übrigens: Die Lichtorgel eignet sich auch prima als Lärmampel für die Schule 😊. Wir können die Funktion auch ohne Sensor testen. Einfach mit den Fingern die beiden Bananenbuchsen (ADC und 3 V) auf der linken Seite berühren. Über unseren Hautwiderstand gibt es eine leitende Verbindung zwischen 3 V und dem analogen Eingang. Dies Phänomen können wir zukünftig z.B. prima für einen Touch Sensor verwenden.

1.2.7 Die Flaschenpost wird flexibel – Hallo Interface

Die vorherigen Kapitel haben sicher zum Nachdenken über eigene Projektideen eingeladen. Zur Umsetzung fehlt jetzt noch die richtige Sensorik und Aktorik. Nachfolgende Tabellen zeigen eine Auswahl an zur Verwendung empfohlener Interface-technik:

Das Internet der Dinge anfassbar machen

Tab. 3: Erweiterungsmöglichkeiten Sensorik

Messgröße	Sensor	Informationen
Luftqualität (Erdbeeren bis Pups)	Air Quality Sensor 1.3	www.seeedstudio.com/wiki/Grove_-_Air_Quality_Sensor_v1.3
Beschleunigung Gravitation Lagebestimmung Vibrationen	9-axis Absolute Orientation Sensor	www.bosch-sensortec.com/bst/products/all_products/bno055
Leitfähigkeit (Erdfeuchte, Hautfeuchte, Wasserstand)	Feuchtig- keitssensor	seeedstudio.com/wiki/Grove_-_Moisture_Sensor
Magnetfeld	HMC5883L 3-Achsen Kompass	seeedstudio.com/wiki/Grove_-_3-Axis_Compass_V1.0
Lautstärke	Loudness Sensor	seeedstudio.com/wiki/Grove_-_Loudness_Sensor
Entfernung (5- 80 cm, Einparksensor)	Abstands- sensor (Sharp)	pololu.com/file/0J85/gp2y0a21yk0f.pdf
Elektr. Strom	Stromsensor (Wechsel- strom)	openenergymonitor.org/emon/buildingblocks/ct-sensors-interface
Elektr. Strom /Spannung	Stromsensor (Gleich- strom)	learn.adafruit.com/adafruit-ina219-current-sensor-breakout
UV-Licht (Sonnenbrand)	Grove - UV Sensor	seeedstudio.com/wiki/Grove_-_UV_Sensor
Bewegung	PIR Motion Sensor	seeedstudio.com/wiki/index.php?title=Twig_-_PIR_Motion_Sensor
Feinstaub	Dust Sensor	seeedstudio.com/wiki/Grove_-_Dust_sensor

38 Das Internet der Dinge anfassbar machen

Tab. 4: Erweiterungsmöglichkeiten Aktorik

Ausgabe	Aktor	Informationen
Elektr.Verbraucher Pumpe, Motor	Relais	seeedstudio.com/wiki/Grove_-_Relay
7x15 Matrix	Charlie- Wing	learn.adafruit.com/adafruit-15x7-7x15-charlieplex-led-matrix-charliewing-featherwing

Das IoT-Kit besitzt je eine analoge und digitale (I2C) Grove Schnittstelle. Sollen mehr Sensoren/Aktoren Verwendung finden, bieten sich folgender Komponenten an:

Tab. 5: Erweiterungen Schnittstellen

Komponenten	Einsatz	Informationen
Adafruit ADS1015	Analog I	learn.adafruit.com/adafruit-4-channel-adc-breakouts/overview
6 Output I/O Expander	Digital	learn.sparkfun.com/tutorials/sx1509-io-expander-breakout-hookup-guide
I2C-Hub	I2C	seeedstudio.com/depot/Grove-I2C-Hub-p-851.html
Sonstiges		
Mobilfunk	GSM	gemalto.com/deutschland/iot
Raspberry Pi	Server	raspberrypi.org/
Bosch Kit	Sensor	xdk.bosch-connectivity.com/

1.3 Für Hacker & Maker

Dieses Kapitel richtet sich an alle technisch vorgebildeten Interessierten, die tiefer in die Schaltung und die flexiblen Möglichkeiten unseres IoT-Kits einsteigen möchten. Weitere Details in der C-Code Version der Blaupausen.

1.3.1 Schaltplan

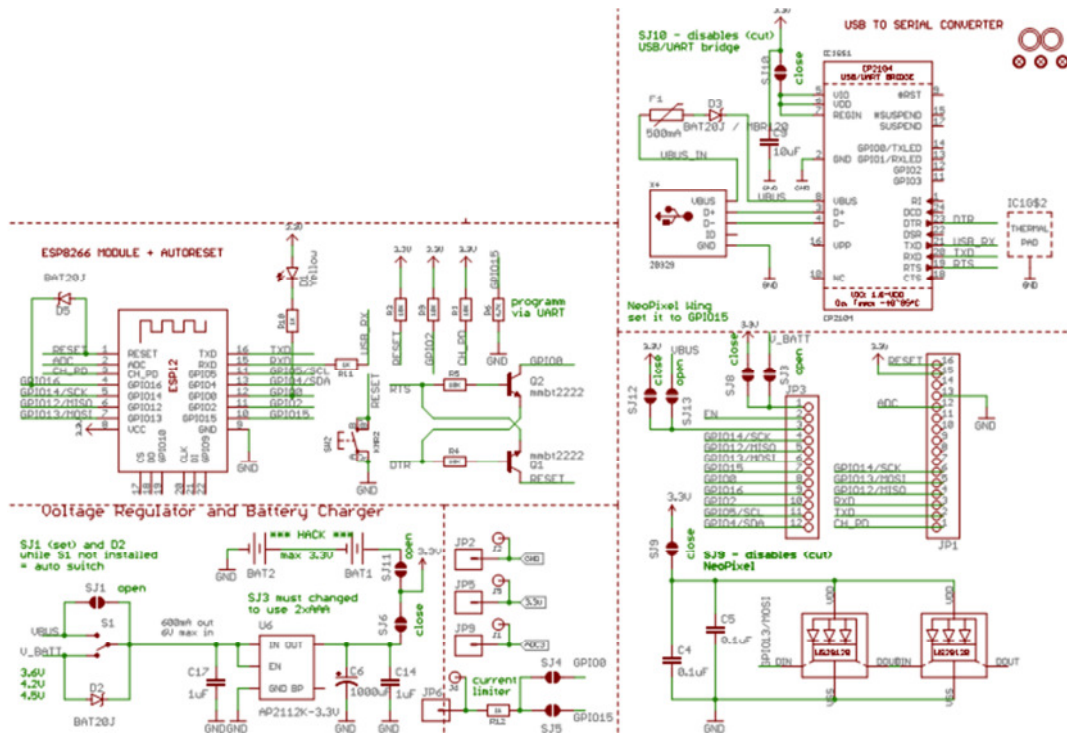


Abb. 27: Schaltplan