

# **Wissens-Appa/Ripa/raturen**

## **Ein epistemologisch-archäologischer Werkstattbericht von der Reparatur eines frühen Mikrocomputers**

---

*Stefan Höltgen und Marius Groth*

Wenn ein defektes Objekt repariert wird, geschieht dabei wesentlich mehr als die Instandsetzung seiner materiellen Qualitäten und Funktionen. Es finden Prozesse der Aneignung, Anwendung und Veränderung von Wissen über das Objekt, seine Funktion und den Reparaturprozess statt. Diese Prozesse beginnen bereits bei der Frage nach dem Defekt (Heidegger 1967: 5), der Vorgehensweise bei der Reparatur, den benötigten Werkzeugen und den Quellen des Reparaturwissens. Während der Reparatur transformiert sich dieses Wissen. Von der Warte einer *Epistemologie des Reparierens* aus stellt das Objekt eine Art Katalysator für Wissenstransformationen dar. Im Rahmen einer Reparatur, die vor allem diesen epistemologischen Charakter fokussiert, wird die Instandsetzung damit zum ›Selbstzweck‹. Der folgende Beitrag folgt diesem Gedanken bei der Reparatur eines historischen Digitalcomputers und stellt die Frage nach dessen Defekt (und wie er gefunden wird), den verwendeten Quellen und Werkzeugen sowie einer spezifischen Kultur der Retrocomputing-Hobbyisten, die sich solcher Reparaturen annehmen. In welchen Beziehungen dies zur Computergeschichte, zu professionellen Bewahrungsstrategien und zu autodidaktischen Methoden des Wissenserwerbs steht, wird dabei ebenso thematisiert werden, wie die spezifische Verfassung des Reparaturguts Computer. Dieser wechselt im Laufe der Reparatur beständig zwischen seinem Status als (zu reparierendem) *Werkgut* und (Reparatur-)*Werkzeug*, um am Ende als Werkzeug schließlich wieder funktional aber unbrauchbar zu sein.

Computer als Gebrauchsgegenstände sind zuvorderst *Werkzeuge* zur Speicherung, Übertragung und Verarbeitung von Informationen. Um diese Funktionen realisieren zu können, bedarf es zweier Voraussetzungen: Computer müssen im Black Boxing ihre technische Komplexität verbergen und auf ein menschliches Maß reduzieren, sodass sie über ihre Schnittstellen (sofort oder nach kurzer Zeit) intuitiv nutzbar sind. Dieses Verbergen hat einen historischen epistemologischen Grund,

der auf John von Neumanns »First Draft of a Report on the EDVAC« (1944) verweist. Dort verbirgt von Neumann die Komplexität der elektronischen Logikgatter hinter dem Begriff »organ«, damit der Ingenieur *den Computer* (nunmehr im Kollektivsingular!) aus einer diagrammatischen Struktur/Architektur entwickeln kann. (von Neumann 1993: 33-36). Aus dieser Sichtweise auf Computer erwächst zudem die Wissenschaft der Computer Science (Informatik), in deren Folge immer weitere »abstraction levels« (Tanenbaum 2005: 5) in den Computer eingezogen werden, um jede von ihnen für Spezialisten differenzierbar und bearbeitbar zu machen, indem die übrigen ausgeblendet werden. Die letzte Eskalationsstufe dieses Prozesses offenbart sich für den Anwender im »ubiquitous computing« (Weiser 1991). Wie in der Fortschrittsgeschichte üblich, geht der Gewinn solcher Spezialisierung und funktionaler Differenzierung mit einem Verlust an allgemeiner Zugänglichkeit und Verstehbarkeit des Ganzen einher (Luhmann 1977).

Die zweite Voraussetzung zum Werkzeuggebrauch von Computern liegt in ihrer *Operativität*. Nur, wenn ein Computer funktioniert (und wenn er eingeschaltet ist), ist er auch als Werkzeug in seinem Sinne nutzbar. Was hier zunächst wie eine Trivialität erscheint, soll im Folgenden von epistemologischer Seite aus diskutiert werden: Was ist ein Computer, der nicht funktioniert? Was bedarf es, um einen defekten Computer zu reparieren? Welche Konsequenzen ergeben sich aus der Reparatur von Computern jenseits ihrer damit (wieder) erreichten Operativität? Und schließlich: In welchem Zusammenhang stehen die beiden genannten Voraussetzungen des Werkzeugcharakters des Computers zueinander?

## COMPUTER ALS HISTORISCHE OBJEKTE

Eine objektfixierte Computergeschichte wird, außer in Historiographien, vor allem in Technik- und Computermuseen vermittelt. Solche Museen widmen sich entweder der Geschichte der Computer im Allgemeinen (z.B. Heinz Nixdorf MuseumsForum, Paderborn) oder bestimmten Technologien (z.B. Analogrechner-Museum, Bad Schwalbach), Epochen (z.B. Oldenburger Computermuseum, Oldenburg), Firmen (z.B. Zuse-Museum, Hoyerswerda), Ingenieuren (z.B. National Museum of Computing, Bletchley/UK) oder Anwendungen (z.B. Computerspielmuseum, Berlin). Neben dieser thematischen Differenzierung lassen sich Computermuseen auch nach dem Grad der Operativität ihrer Exponate unterscheiden. Die Skala reicht hier von der vollständig dysfunktionalen Hardware-Ausstellung (z.B. Computermuseum Kiel) bis zum Hands-on-Museum (z.B. Oldenburger Computermuseum, Oldenburg oder Living Computer Museum, Seattle). Letztere akzentuieren Computer in ihren Ausstellungen als Apparate im oben genannten Sinne (zum Speichern, Übertragen und Verarbeiten von Informationen) und vermitteln ihren Besuchern genau diese Aspekte ihrer historischen Genese. Dabei verursachen sie eine produktive Spannung zwischen Computern als historischen Artefakten und operativen Medien, die

notwendig gegenwärtig (Ernst 2012: 42) sein müssen: Ein eingeschalteter historischer Computer befindet sich radikal im Hier und Jetzt. Erst so kann er aber seine technische ›Historizität‹ unter Beweis stellen: Verarbeitungszeiten, Ladezeiten, Funktionsbeschränkungen, spezifische Schnittstellen usw. früherer Computertechnologie im Vergleich zu heutiger lassen sich nur auf diese Weise erfahrbar machen.

Neben Museen existieren noch weitere ›Orte‹, in denen Computergeschichte erfahrbar wird. Dies sind vor allem private Sammlungen, Vereine und Szene-Treffs wie Börsen, Flohmärkte und Festivals. Hier ist das Verhältnis von dysfunktionalen zu funktionalen historischen Computern zugunsten letzterer besonders deutlich: Oftmals ist der ›Grund‹ einer Sammlung, eines Verkaufs oder einer Ausstellung die Erfahrbarmachung von Computergeschichte am operativen Gerät – etwa, um damit historische Computerspiele spielbar zu machen, kreative Wettbewerbe zu inszenieren oder Hardware- und Software-Neuentwicklungen für alte Plattformen vorzuführen. Da es im Wesen ›privater‹ Computersammler liegt, aus ihrem Hobby eben keine Profession zu machen (oder anders herum: zumeist nicht zugleich professioneller Sammler/Kurator *und* Hobbyist zu sein), stehen diese oftmals vor dem Anspruch, ihre Sammlungsgegenstände lauffähig zu halten oder wieder lauffähig zu bekommen, ohne sich hierzu einer professionellen Infrastruktur bedienen zu können (Takhteyev/DuPont 2013: 432). »Bewahrung«<sup>1</sup> findet hier in der Regel auf Basis privater Mittel und mit autodidaktisch erworbenem Wissen statt, das insbesondere an den konkreten Plattformen, die zur Sammlung gehören, spezialisiert wird. Die dafür erworbenen Kenntnisse und Praktiken gehören immer schon zum Inventar des privaten Mikrocomputer-Nutzers (Zaks 1981).

Das Signallabor des Instituts für Musikwissenschaft und Medienwissenschaft an der Berliner Humboldt-Universität ist diesbezüglich ein ›hybrider‹ Ort, da dessen Ausstattung und Funktion zwischen diesen beiden Sphären situiert ist. *Professionell* ist es insofern, dass es sich bei den dort gesammelten Mikrocomputern um Objekte einer akademischen Sammlung handelt, die Gegenstand eines medienwissenschaftlichen Forschungsprojektes sind. *Privat* ist diese Sammlung insofern, als die Computer nahezu sämtlich aus privaten Mitteln des Projektinhabers angeschafft wurden und dadurch, dass dieser, wie die oben beschriebenen Hobbyisten, als Medienwissenschaftler keine professionellen/akademischen Kenntnisse in Elektronik besitzt, um diese bei der Instandhaltung und -setzung anwenden zu können; er ist

1 Der Begriff der *Bewahrung* (engl. *preservation*) ist hier eng mit dem Thema der *Restaurierung* verwandt. Bei letzterem handelt es sich um eine komplexe Praxis, die einen Zustand des Artefaktes herstellen soll, der weitgehend einem Originalzustand entspricht. Restaurierte Objekte unterscheiden sich von reparierten dadurch, dass Funktionalität kein notwendiges Ziel darstellt. Denn ein zentrales Anliegen des Restaurierens ist es, den materiellen Zustand des Objektes nicht durch neue Bestandteile zu ergänzen oder zu ersetzen, um beispielsweise Gebrauchsspuren und Verwendungswiesen des Objektes sichtbar zu halten. Demgegenüber verlangt die *Reparatur* historischer Objekte nicht selten die Verwendung moderner Ersatzteile und stellt damit (historische) Hybride her (Burnet/Supnik 2006; Takhteyev/DuPont 2013).

jedoch innerhalb der Retrocomputing-Szenen gut vernetzt, um Wissen ›rekrutieren‹ zu können. Dies ist zudem ein weiterer Forschungsaspekt, in dem es darum geht, zu ermitteln, auf welche Weise autodidaktisches Wissen über Computertechnik und -geschichte in den sogenannten Retrocomputing-Szenen erworben, eingesetzt und weitergegeben wird.

## COMPUTERARCHÄOLOGIE

Im Hintergrund des Forschungsprojektes steht eine kritische Auseinandersetzung mit der Computergeschichtsschreibung als Erweiterung der Medienarchäologie.<sup>2</sup> Hierbei werden folgende drei Aspekte berührt: Ausgehend von der oben gemachten Feststellung, dass Computer durch ihre Operativität definiert sind, stellen sie erstens einen für die Geschichtsschreibung problematischen Gegenstand dar. Ein operativer ›alter‹ Computer ist nicht historisch, sondern gegenwärtig. Dies besitzt auch ganz praktische Relevanz: Musealisierte Computer müssen bewahrt werden; ihre Instandsetzung und Inbetriebnahme können dieses Sammlungsziel gefährden, weil sie das historische Artefakt verändern (Takhteyev/DuPont 2013). Während Musealisierung auf Bewahrung eines historischen Ist-Zustandes abzielt, verfährt Computerarchäologie »redaktionell« (Ernst 2004: 253f.) verändernd, um historische und kontemporäre Elemente miteinander in Konflikt zu bringen. Dies lässt sich aber nur am operativen Objekt vollziehen. Defekt und abgeschaltet verbergen diese Artefakte jedoch ihren Charakter als Werkzeuge: Sie sind keine Computer, sondern Hardware. Zweitens unterliegt die Geschichtsschreibung des Computers denselben Fiktionalisierungsprozessen (White 1994) und Machtdispositiven (Foucault 2003: 392) wie die Historiographie anderer Gegenstände. Bei Computern kommt allerdings hinzu, dass sie als technische Objekte nicht mehr allein durch historiographische Diskurse beschreibbar sind, sondern dazu *technomathematische Methoden* verwenden und mithin interdisziplinäre Perspektiven eingenommen werden müssen. Diagramme, Codes und Schaltpläne sind der hermeneutischen Lektüre des Historikers jedoch unzugänglich. Drittens kann die Beschreibung eines Computers als

---

2 Medienarchäologie ist eine ab Mitte der 1980er Jahre von Friedrich Kittler, Siegfried Zielinski, Wolfgang Ernst und anderen Medienwissenschaftlern entwickelte Theorie und Methode zur Erforschung des medientechnischen Aprioris von Wissen. Da die Grundannahme hierzu lautet, dass Medientechnologien wesentlichen Anteil an der Form(at)ierung von Medieninhalten und damit dem durch sie vermittelten Wissen haben, untersucht Medienarchäologie mit technomathematischen Methoden Medien in ihrer Genese und Wirkung auf das Wissen. Im Sinne der Archäologie Foucaults fokussiert Medienarchäologie besonders die Brüche und Anachronismen der Medien geschichtsschreibung und sieht in der notwendigen Operativität von Medien (die im Vollzug sein müssen, um im Medienzustand zu sein) einen Widerspruch zu deren Historisierung. Ein historisches Radio ist heute – als Medium in Vollzug gesetzt – nicht mehr historisch, sondern radikal gegenwärtig (Ernst 2012).

technikhistorisches Artefakt überdies nicht mehr rein diskursiv erfolgen, sondern erfordert auch die *Demonstration* seiner Operativität, um beispielsweise Thesen historischer Provenienz belegen zu können. Die Komplexität von Computerprozessen, etwa beim Ablauf eines Spielprogramms, lässt sich nicht allein aus Codes, Schaltplänen und Diagrammen ermitteln; sie zeigt sich erst als Emergenz der zusammenwirkenden Prozesse in Hard- und Software am laufenden System. Die *Demonstration des Computers* ergänzt zusätzlich den *Diskurs über den Computer*, der als »Archäografie« (Ernst 2012: 385) und/oder »technical report« (Montfort 2013) bereits nicht-symbolische und nicht-diskursive Aspekte wie Schaltpläne, Diagramme, mathematische Beschreibungen, Messwertreihen usw. notiert. Sie ermöglicht es – ähnlich wie eine Inbetriebnahme alter Geräte oder die Rekonstruktionen von historischen Instrumenten –, den rein »beschreibenden« Diskurs mit all seine Effekten (Fiktionalisierung, Macht, usw.) zu umgehen.

Diese sehr knappe Darstellung dessen, was als Theorie der Computerarchäologie entwickelt wird, korrespondiert auf der praktischen Seite mit einer alternativen Computergeschichtsschreibung, die aus genannten Gründen nicht mehr allein durch (machtgesteuerte) Diskurse (etwa in publizierten Monographien, in Fernsehsendungen usw.) realisiert werden kann, sondern notwendigerweise stets den Widerspruch, den alternativen Blick, die Quellen-Korrektur, die Konzentration auf die Brüche und Diskontinuitäten – zur Erwiderung auf die »story types« (White 1994: 135) und auf die großen Erzählungen von ›wirtschaftlichem Erfolg‹, ›Fortschritt/Progression‹, ›Genies/Ingenieuren‹ usw. hinter den Historiographien – formulieren muss. Diese Arbeit wird für die Weiterentwicklung der Computergeschichte bislang vor allem durch die Hobbyisten übernommen.

Das Gegenstandswissen der Hobbyisten rekrutiert sich aus unterschiedlichsten Quellen: Bücher, Filme, Zeitschriften, soziale Netzwerke im Internet, Szene-treffen, Ausstellungen und andere Quellen werden (im akademisch-historischen Sinne) ›unprofessionell‹ zu einem historischen Wissen über Computer amalgamiert, zu dem das eng an den Geräten orientierte technische Wissen hinzutritt. Retrocomputing-Hobbyisten kennen die von ihnen gesammelten Plattformen oft besser als die Ingenieure, die sie in der Vergangenheit entworfen haben – nicht nur, weil sie den historischen Rückblick besitzen, sondern auch, weil sie auf vielfältige und langjährige Experimente mit den Rechnern zurückblicken können, in denen sie undokumentierte Funktionen, Design Flaws und anderes entdeckt, beschrieben und praktisch ausgenutzt haben. Dieses zweite Wissen kann nur in der konkreten Auseinandersetzung mit den operativen Systemen entstehen, die ihnen hierbei zugleich als *Werkzeug und Gegenstand ihres Wissenwollens* dienen: Wenn Computer in Maschinensprache programmiert werden, sind sie zunächst ein Programmierwerkzeug; wenn durch diese Programmierung eine nicht-dokumentierte Funktion exploriert wird, werden sie zum Gegenstand der Auseinandersetzung. Computerarchäologie als *Wissenspraxis* untersucht diesen spielerischen Zugang zu

Computerwissen. Der unten beschriebene Test des Speichertestprogramms mittels bewusst hervorgerufenem Fehler wird diesen dialogischen Lernprozess zeigen.

Wie eingangs geschrieben, stellt die Reparatur eines historischen medientechnischen Objektes für die Computerarchäologie mehr als bloß dessen Instandsetzung zur Wiedererlangung einer Operativität dar. Der Prozess des Reparierens ist eine intellektuelle und manuelle Form der Auseinandersetzung mit dem Objekt, die gänzlich andere Methoden und Werkzeuge als dessen historische und diskursive Beschreibung erfordert. In der Auseinandersetzung mit dem konkreten medientechnischen Objekt, dem ‚hands-on‘, begegnet der Medienwissenschaftler seinem Forschungsgegenstand auf seiner ihm eigenen phänomenalen Ebene. Die ›Unmenschlichkeit‹ des Technischen zeigt sich hier in all ihrer Deutlichkeit: Werden Computer auf diskursiver Ebene z.B. als »Digitalmedien« dargestellt, die »nur 0 und 1 kennen«, die als »Denkmaschinen« genutzt werden, die einen fiskalisch messbaren »Wert« besitzen, mit denen »gehackt« oder »Computerkunst« geschaffen wird usw. – die also eine Geschichte, Soziologie, Ästhetik, Politik, Philosophie haben und dadurch Bezüge zur menschlichen Kultur besitzen, so zeigen sie sich dem ›kalten Blick‹ (Ernst 2004: 244) des Archäologen ausschließlich als Schaltungen, die auf Basis von physikalischen und elektronischen Gesetzmäßigkeiten Signale verarbeiten. Alle genannten kulturellen Effekte finden sich hier (noch) nicht, sondern ergeben sich erst als Emergenz- und Nutzungseffekte aus dieser technischen Verfasstheit. Diese Tatsache zu berücksichtigen, stellt ein zentrales Anliegen computerarchäologischer Forschung dar, denn in diesem technischen Apriori (Ernst 2004: 241) gründet sich letztlich der spezifische Charakter der Computerkultur. Während Museen Computer(hardware) als sichtbare Indizes solcher historischen, kulturellen, ästhetischen und anderen Diskurse sammeln und zeigen, fokussiert Computerarchäologie Computer als Apparate, die – losgelöst von diesen übergeordneten Sichtweisen – zunächst ›für sich‹ Forschungsgegenstände darstellen.

Der nachfolgende Werkstattbericht beschreibt die Applikation dieser Theorie und die Anwendung der Methoden anhand der Reparatur eines defekten Mikrocomputers. Dabei handelt es sich um den Sol-20, der weithin als ›einer der ersten Homecomputer‹ gilt – also als frei verkäuflicher Mikrocomputer, bei dem alle zum Betrieb benötigten Technologien (Computer, Tastatur, I/O-Schnittstellen) in einem Gehäuse verbaut sind, und der Anschlussmöglichkeiten für privat verfügbare Medientechnologien (Kassettenrecorder, Fernseher, Schreibmaschine, Telefon) besitzt. Der Sol-20 wurde 1976/77 vorgestellt und ist selbst das Ergebnis eines Projektes, das in die oben genannte Retrocomputing-Szene passen würde: Er entstand als Bausatz für frühe Computerhobbyisten (vgl. Abb. 1) und als Erweiterung der Mitte der 1970er Jahre populären TV-Typewriter, mit denen man selbst eingegebene Schrift auf den heimischen Fernseher darstellen konnte (Levy 2010: 184f.). Lee Felsenstein, der federführende Ingenieur hinter dem Sol-20, stammt aus einer Tradition von Hackern, die sich bereits in den späten 1960er Jahren die Popularisierung und Demokratisierung von Computertechnologie auf die Agenda geschrieben

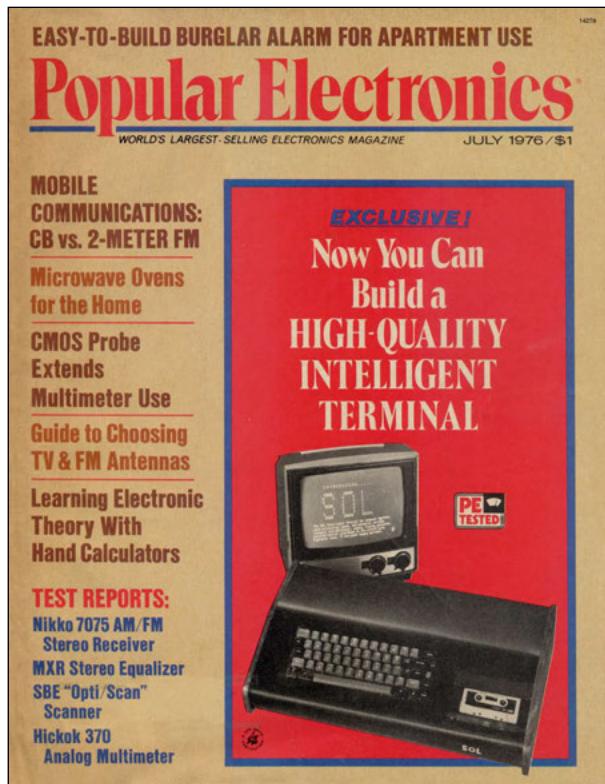


Abb. 1: Cover der Zeitschrift Popular Electronics, in der der Sol-20 im Juli 1976 als Bausatz vorgestellt wurde.

hatten (Höltgen 2014). Der Finanzier des Projektes, Les Solomon, gab etwa zur Zeit der Entstehung des Sol-20 eine Zeitschrift für Computerhobbyisten heraus (»Creative Computing«), in der die ersten Computerbausätze vorgestellt wurden und die als Wegbereiter der privaten Computerkultur gesehen wurde (Roberts/Yates 1975).

Vom Sol-20 (vgl. Abb. 2) sind mutmaßlich 10.000 Exemplare (Battle 2006) gebaut und entweder als Bausatz oder Fertigerät vertrieben worden. Aufgrund der schon damals rasanten Entwicklung der Mikrocomputertechnologie sowie der spezifischen Fehleranfälligkeit des Sol-20 (siehe unten) hat diese vergleichsweise geringe Zahl dazu geführt, dass heute nur noch wenige Exemplare existieren dürfen. Sein technischer Aufbau ist typisch für Mikrocomputer seiner Entstehungszeit. Er basiert auf dem 8-Bit-Mikroprozessor 8080 der Firma Intel, der mit 2 Megahertz getaktet ist. Als 8-Bit-System kann er maximal 64 Kilobyte RAM verwalten; ausgeliefert wurde er zumeist mit 16 Kilobyte, die über Steckkarten erweitert werden konnten. Zur Erweiterung wurde das Bus-System S-100 (vgl. Abb. 3



Abb. 2: Der Sol-20 aus der Sammlung des Münchener Computerhistorikers Hans Franke (defekt, unmodifiziert).

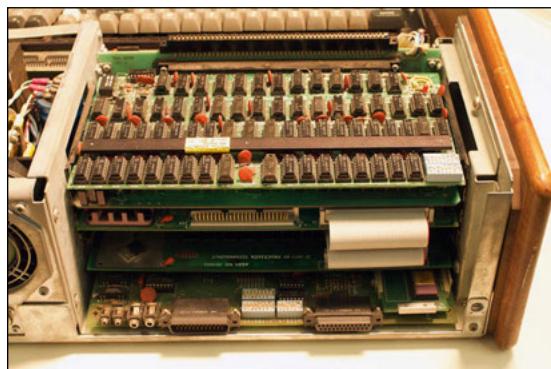


Abb. 3: Das S-100-Bus-System im Sol-20 voll bestückt mit Erweiterungskarten.

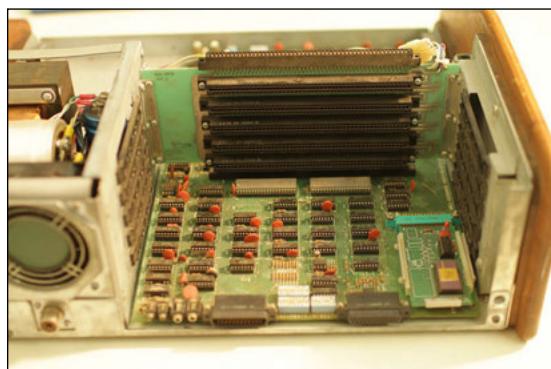


Abb. 4: Das S-100-Bus-System im Sol-20 ohne eingesteckte Karten.

u. 4) genutzt, das bei vielen zeitgenössischen Mikrocomputern zum Einsatz kam und damit eine frühe Form eines Standards bildete. Für dieses Bus-System gab es zahlreiche Erweiterungskarten zu kaufen und die Anwender konnten zudem leicht eigene Steckkarten entwickeln, womit der Sol-20 einen ›Bastel-Computer‹ darstellte, der den Bedürfnissen seiner frühen Nutzerclientel (Hobby-Elektroniker) sehr entgegen kam. Wie geschildert, integrierte der Sol-20 allerdings bereits Elemente, die für andere damalige Systeme erst dazu gekauft oder gebaut werden mussten: eine Tastatur, eine Videobild-Erzeugung und Ein-/Ausgabeschnittstellen für Massenspeicher (Disketten- und Kassettenlaufwerk), Drucker, serielle Peripherie. Damit genügte der Sol-20 ebenso einer Nutzergruppe, die weniger auf Basteln, sondern auf Anwendungen orientiert war. Für letztere entstand sukzessive ein Angebot an Amateur- und professioneller Software verschiedener Gattungen. Durch die Möglichkeit, auf dem Sol-20 das seinerzeit weit verbreitete Betriebssystem CP/M zu nutzen, erweiterte sich der Einsatzbereich für professionelle Anwendungen immens: Büroanwendungen, Programmiersprachen, Lernsoftware und vieles andere konnte über CP/M auf dem Computer genutzt werden.

Der Sol-20 stellt heute ein vergleichsweise seltenes Objekt dar. Dies liegt zum einen an der zur Zeit seiner Entstehung hohen Innovationsfrequenz: Immer neue, verbesserte, schnellere und günstigere Computer erreichten den Markt; für alte Geräte gab es kaum Verwendung, weshalb sie oft entsorgt oder demontiert wurden. (Letzteres ist bei einem auf einem Standard-Bussystem<sup>3</sup> wie dem des S-100 basierenden Systems sehr wahrscheinlich, weil viele Erweiterungen kompatibel zu nachfolgenden S-100-Computern waren.) Seine Verarbeitung und insbesondere die Verwendung von organischem Material – Holz für die Seitenleisten und organische Schaumstoffe für die Tastatur – ließen den Computer zudem schneller an Zerfallsprozessen leiden. Aufgrund dieser Beschaffenheit dürfte heute kein funktionierender Sol-20 mit allen ursprünglichen Materialien bzw. Bauelementen mehr existieren. Seine Reparatur stellt mithin die einzige Möglichkeit dar, den Sol-20 als Computer (im hier verstandenen Sinne) erfahrbar zu machen.

## WERKSTATTBERICHT

Erhalten haben wir den Sol-20 im Mai 2015 als Dauerleihgabe auf dem Vintage Computer Festival Europa von Hans Franke, einem Sammler aus München. Die Prämisse war, dass der Rechner repariert und im Signallabor für das Forschungsprojekt genutzt werden sollte. Ursprünglich war geplant, den Rechner auf dem Vintage Computing Festival Berlin im Herbst 2015 von seinem Erbauer, Lee

3 Der Standard des S-100-Bussystems ist unter IEEE 696-1983 definiert: [http://www.imsai.net/download/IEEE\\_696\\_1983.pdf](http://www.imsai.net/download/IEEE_696_1983.pdf) (zuletzt abgerufen 27.07.2017).

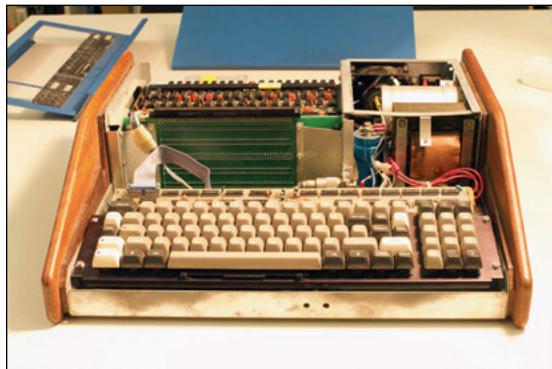


Abb. 5: Der geöffnete Sol-20.



Abb. 6: Die Tastatur mit abgezogenen Tastenkappen zeigt die zerfallenen Schaumstoffpads. Die weiße Substanz zwischen den Tasten (Bildmitte) ist ein verlassener Insektenkokon.

Felsenstein, vor Ort reparieren zu lassen, was dann leider wegen Terminüberschneidungen nicht möglich war.

Aufgrund der Aussage des Besitzers, dass der Computer generell funktioniere, verlief die Vorabkontrolle vor seiner ersten Inbetriebnahme recht spartanisch: Das Gehäuse wurde geöffnet und das ‚Innenleben‘ begutachtet (vgl. Abb. 5). Zu unserer Überraschung waren alle ICs bereits gesockelt, sodass etwaige Fehler im digitalen Bereich mit geringem Aufwand zu beheben sind, da die ICs leicht aus den Sockeln gezogen werden können, anstatt sie zeitaufwändig von der Platine entlöten zu müssen, was immer mit dem Risiko der Beschädigung der Leiterplatte einhergeht. So sehr dies eigentlich eine spätere Erleichterung darstellt, so werden diese Sockel allerdings im Laufe der Zeit auch zur potenziellen Fehlerquelle, da dort durch Oxidation Kontaktenschwierigkeiten entstehen können. Außerdem sind große Teile des Rechners auf Zusatzkarten ausgeführt (2 mal 16 Kilobyte RAM, 2 Schnittstellen-

karten, »Personality Module« mit Betriebssystem); auch diese können von Oxidation befallen sein. Also wurden alle Steckkarten gezogen und die Kontakte vorsorglich mit Isopropanol gereinigt. Danach wurden mehrere Steckzyklen durchgeführt. Im Anschluss wurden alle ICs auf dem Mainboard von Hand nachgedrückt. Zusätzlich wurde mit einem Pinsel der Staub vom Mainboard gebürstet. Daraufhin erfolgte eine erste Messung mit einem Multimeter, um auszuschließen, dass auf der Platinen Kurzschlüsse vorhanden sind. Der Test ergab keine derartigen Probleme. Staub und Verschmutzungen wurden oberflächlich von den Platinen entfernt. Unter zwei Tastenkappen fanden sich leere, verpuppte Insektenkokons, die entfernt wurden.<sup>4</sup>

Es sprach also nichts gegen eine erste Inbetriebnahme. Da der Computer ausschließlich in den Vereinigten Staaten von Amerika produziert und vertrieben wurde, war es notwendig, die hiesige 230-Volt-Netzspannung mittels Step-Down-Converter auf 110 Volt anzupassen. Zu diesem Zeitpunkt bestand die Hoffnung, dass der Rechner aus der Netzfrequenz keinen Takt ableitet, denn diese unterscheidet sich bei deutscher (50 Hertz) und US-amerikanischer (60 Hertz) Wechselspannung. Als Monitor stand ein Gerät zur Verfügung, dass das NTSC-Bildformat akzeptiert. Das erste Einschalten ergab lediglich wirre Zeichen auf dem Bildschirm, wobei es sich vielleicht nur um ›Anlaufschwierigkeiten‹ handelte. Ein *power cycle* brachte dann tatsächlich den zu erwartenden Cursor auf den Bildschirm. Eingaben via Tastatur waren allerdings nicht möglich. Im eingeschalteten Zustand wurden die Spannungen im Rechner kontrolliert; diese lagen alle im Soll-Bereich.

## Reparatur der Tastatur

Der nächste Schritt war die Suche nach den Schaltplänen des Sol-20. Der erste Treffer einer Google-Suche führte zur Internetseite von Jim Battle, wo nicht nur Schaltpläne, sondern auch Anleitungen, Programme, typische Fehler und Reparaturen, ein Emulator und anderes Wissen zum System abrufbar waren (Battle 2006). Dort fand sich auch der Hinweis, dass ein typisches Problem, welches inzwischen bei nahe jeden Sol-20 betrifft, die Tastatur ist. Diese ist in ihrer Bauart besonders: Sie verwendet nämlich keine Mikrotaster, welche einen Schaltkreis schließen, sondern mit einem leitfähigen Material bedampfte Plastikscheiben, die mittels Schaumstoff an den Tastenkappen angebracht sind und somit die Kapazität des Schaltkreises verändern, sobald man sie senkt oder hebt – eine Technik, von der wir bis zu diesem Punkt nicht wussten, dass sie auch im Computerbereich existierte. Diese Bauart wurde recht schnell vom Markt verdrängt, erleichterte aber unseren weiteren Reparaturverlauf sehr. Als Fehlerquelle konnten mit sehr hoher Wahrscheinlichkeit die Schaumstoffpads identifiziert werden, welche im Laufe der Jahre spröde wurden und dann im Wortsinne ›zerbröselten‹ (vgl. Abb. 6), wonach die Plastikscheiben

4 Nicht ohne dabei an Grace Hoppers berüchtigten Labortagebuch-Eintrag »First actual case of bug being found« zu denken (Hopper 1947).

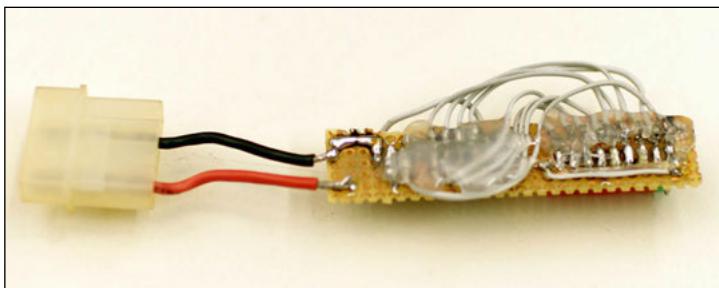


Abb. 7: Selbstgebauter Adapter zum Testen der Tastatur-Funktionen.

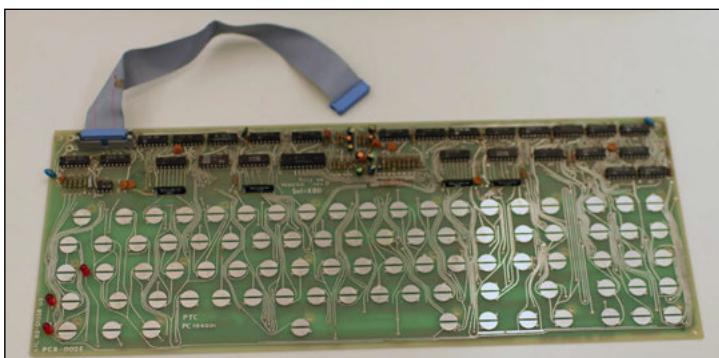


Abb. 8: Gereinigtes Tastaturlfeld vor dem Bekleben mit neuen Schaumstoffpads.

dauerhaft auf der Tastaturplatine aufliegen und die Tasten damit aus elektronischer Sicht für den Rechner ›klemmen‹. Die exotische Technik der Kapazitätsänderung zum Registrieren von Tastendrücken stellte sich für uns an diesem Punkt als vorteilhaft heraus: Die ›nackte‹ Tastaturplatine ohne Tastenmechanik lässt sich nämlich prinzipiell auch mit den Fingern bedienen (wenn auch nicht ganz ohne Probleme – wie weiter unten gezeigt wird).

Zusätzlich zu dieser Fehlerbeschreibung fand sich noch die Website von Herb Johnson, auf welcher die Anleitung für den Bau eines Tastatur-Testers zu finden war (Johnson 2012). So konnte die Funktion der Tastatur überprüft werden – unabhängig vom Rechner und ohne sie komplett reparieren zu müssen. Dieser Adapter (vgl. Abb. 7) wurde mit vorhandenen Bauteilen schnell auf einer Lochrasterplatine nachgebaut. Daraufhin wurde die Tastatur von ihren Tasten befreit. Ein Blick auf die Schaumstoffpads ergab, was Battle auf seiner Webseite geschildert hatte: Diese hatten sich im Laufe der Zeit nahezu vollständig aufgelöst und zusätzlich die beschichteten Plastikplättchen angegriffen. Mit dem Tastatur-Adapter ließ sich recht schnell feststellen, ob die Elektronik der Tastatur funktioniert. ›Tastendrücke‹, welche mit den nackten Fingern ausgelöst werden konnten, wurden korrekt auf dem

Test-Adapter angezeigt. Dieser zeigte die unterschiedlichen 8-Bit-Muster für die Zeichen sowie das Strobe-Signal zum Einlesen des Musters mittels Leuchtdioden an. In der Hoffnung, dass sich der Rechner mit der ›nackten‹ Tastaturplatine (vgl. Abb. 8) bedienen lässt, wurde diese wieder angeschlossen – jedoch vergebens: Es wurden zwar Zeichen entgegengenommen, allerdings prellten diese so stark, dass anstatt einzelner Symbole quasi zufällige Zeichenketten von der Tastatur an den Rechner übertragen wurden. Damit ließ sich allerdings bereits ein großer Teil der Reparatur abhaken – elektronisch schien der Rechner das zu tun, was er sollte. Lediglich die Auslösemechanik der Tastatur war reparaturbedürftig.

Auf den beiden vorher benannten Internetseiten wurde dieses Thema ausführlich behandelt. Die originalen Schaumstoffpads und zugehörigen Plättchen werden seit circa 15 Jahren nicht mehr kommerziell hergestellt. Es gibt aber Möglichkeiten diese nachzubauen. Dazu wird Schaumstoff mit doppelseitigem Klebeband auf beiden Seiten ausgestanzt, auf der oberen Seite ein Plastikplättchen aufgeklebt, welches von der Tastenmechanik gehalten wird, und die Unterseite, die Kontakt mit der Tastaturplatine hat, mit Mylarfolie versehen. Vorgefertigte Schaumstoffpads gab es auf eBay von einem Dritthersteller zu kaufen,<sup>5</sup> sodass ein Ausstanzen durch uns entfiel. Für den Ersatz der Mylarfolie gab es mehrere Optionen: Innenseiten von Kartoffelchipstüten, Heliumballons aus Geschenkshops oder Erste-Hilfe-Decken wurden als potenzielle Quellen benannt. Die Entscheidung fiel auf Mylarfolie für Gewächshäuser.

Nach dem Eintreffen der Schaumstoffpads und der Mylarfolie sollte die Reparatur fortgesetzt werden; die Mylarfolie erwies sich jedoch als unbrauchbar. Sie war nämlich mit einer zusätzlichen Plasticschicht versehen, sodass sie nicht leitfähig war, wie ein Test mit dem Multimeter ergab. Ein Entfernen dieser Schicht war nicht möglich. Zum Test wurde dennoch eine einzelne Taste damit versehen. Dazu wurden die alten Schaumstoffreste, Plättchen und die Folie aus einer Taste entfernt, das Plättchen von Klebe- und Schaumstoffresten befreit und ein neues ›Sandwich‹ aus Plastikplättchen, Schaumstoff und Mylarfolie gebaut – ein Arbeitsschritt, der pro Taste circa drei bis fünf Minuten in Anspruch nahm, da einerseits das Entfernen der Plastikplättchen kompliziert war und zum anderen die Mylarfolie manuell auf das richtige Format zurechtgeschnitten werden musste. Die gesamte Tastatur des Sol-20 besitzt 85 Tasten. Daher fand der Test zunächst mit nur einer Taste statt. Wie erwartet, erbrachte er keine Funktion. Daher wurde nun nach anderen, vorhandenen Materialien Ausschau gehalten. Ein erster Test mit Alufolie erwies sich als nicht zufriedenstellend, da die Tasten, ähnlich wie mit den nackten Fingern, stark prellten – also nicht korrekt auslösten und somit Zufallszeichen produzierten. Die Wahl fiel schließlich auf etwas, was per Zufall in den Blick kam, aber gute Ergebnisse lieferte: elektrosensitives Metallschicht-Papier aus einem Drucker für einen anderen historischen Computer des Signallabors, welches in ausreichenden

5 <http://www.ebay.de/itm/121266887970> (zuletzt abgerufen 27.07.2017).

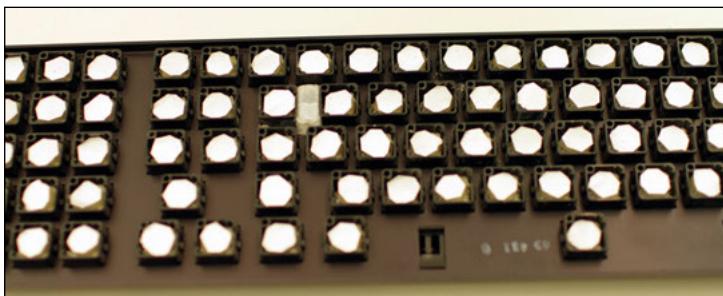


Abb. 9: Die mit neuen Schaumstoffpads und Kontaktfolie (Metallschicht-Papier) versehene Tastatur.

Mengen verfügbar war. Dieses Papier war überdies auch leichter zuschneidbar als Mylar- oder Alufolie. Die komplette Reparatur der Tastaturstempel dauerte circa vier Stunden. Als Ergebnis (vgl. Abb. 9) besaß der Sol-20 danach eine >recht gut-funktionierende Tastatur. >Recht gut< deshalb, weil einige wenige Tasten immer noch prellten. Diese prellenden Tastenstempel wurden dann auf den Zehner-Ziffernblock umgelegt, da dessen Tasten redundant sind (alle Ziffern befinden sich auch oberhalb der Buchstabenreihe).

## RAM

Der Sol-20 war nun funktionstüchtig und konnte wieder betrieben werden. Ein Blick in das sehr detaillierte Handbuch, welches auch als Aufbauanleitung diente – der Rechner wurde, wie geschrieben, sowohl als Fertigerät als auch als Bausatz vertrieben –, offenbarte die Möglichkeit einzelne Bytes in das RAM zu schreiben und wieder auszulesen, was auch funktionierte. Der nächste Schritt war nun, Software in den Rechner zu laden.

Dazu waren ab Werk drei Möglichkeiten vorgesehen: Programme manuell über das implementierte Monitorprogramm einzugeben, was sehr mühselig (übliche Programme sind circa zwei DIN-A4-Seiten lang) und auch sehr fehleranfällig (es handelt sich dabei um reine Zahlenfolgen ohne automatisierte Korrektheitskontrolle) ist. Alternativ kann man Programme, welche sich als Listing auf einem heute üblichen PC befinden, über die serielle Schnittstelle in den Sol-20 übertragen. Leider ist aber diese Schnittstelle heute kaum noch in Computern vorhanden<sup>6</sup> und hat obendrein mehrere Standards durchlaufen (Übertragungsraten, Verdrahtungsschema, Spannungen), sodass dies im Fehlerfall sogar den Sol-20 beschädigen könnte. Die Entscheidung fiel auf die dritte, typische >damalige< Variante, Software über

---

6 Es existieren Adapter-Lösungen, um den USB-Port als RS-232-kompatible Schnittstelle zu nutzen.

den Kassettenport des Computers einzuspielen – wenn auch mit modernen Hilfsmitteln per Handy und Sounddatei. Ein Sol-20-User mit dem Nicknamen »Lord Philip« hatte bereits einen Großteil der verfügbaren Software mit den vorher erwähnten Methoden in seinen Sol-20 eingetippt bzw. überspielt und die Programme als .wav-Sounddateien abgespeichert und auf seiner Internetseite zur Verfügung gestellt (Anonymous o.J.).

Anstatt eines Kassettenrekorders wurden ein Smartphone und später ein MP3-Recorder mit SD-Karte als Audio-Abspielgerät verwendet, da auf Kassetten ohnehin lediglich analoge Töne abgespeichert sind, welche erst im Rechner digitalisiert werden. Dieses Prinzip der Übertragung von Software über das Smartphone konnte bereits bei anderen Computern aus den 1980er Jahren mehrfach erfolgreich angewendet werden – wie nun auch beim Sol-20. Der Großteil der so in den Computer geladenen Programme funktionierte – lediglich der BASIC-Interpreter von Microsoft führte zu der Fehlermeldung, dass zu wenig Speicher verfügbar war – obwohl mit 32 Kilobyte eigentlich ausreichend RAM eingebaut war. Irgendetwas schien also noch nicht zu funktionieren. Eine kurze Verifikation mittels Emulator (Battle 2005) bestätigte, dass der BASIC-Interpreter mit 32 Kilobyte RAM funktionieren sollte. Der Sol-20 lag nun zunächst in einer funktionierenden 16-Kilobyte-Variante vor. Die zweite 16-Kilobyte-RAM-Karte hatten wir zur Sicherheit aus dem Slot gezogen, um so möglichen Konflikten vorzubeugen. Einschränkungen waren dadurch keine feststellbar.

Durch das eingebaute Monitorprogramm war es zwar möglich, Werte in den RAM-Bereich der Speichererweiterung zu schreiben und auch wieder erfolgreich auszulesen. Die Karte schien also prinzipiell zu funktionieren. Ein obligatorischer »Fingertest« – fühlen, ob sich einzelne RAM-Chips unverhältnismäßig stark erhitzten – ergab keine Auffälligkeiten. Ein Auslesen des kompletten RAM-Bereichs erbrachte ein regelmäßiges Muster, was dem normalen Initialzustand von dynamischem RAM entspricht (32 mal &00, 32 mal &FF im Wechsel). Es war also davon auszugehen, dass kein Defekt auf der Steckkarte vorliegt, sondern dass das Problem woanders zu suchen ist.

Ein Blick auf die Karte selbst, sowie in die Anleitungen zum Sol-20 und zur Karte offenbarte ein »triviales« Problem: RAM auf Zusatzkarten muss im Sol-20 in einem durchgehenden Block liegen. Eingestellt war, dass Karte 1 die Adressen 0 bis 16384 enthält und Karte 2 Adressen 32768 bis 49152. Ein kompletter 16-Kilobyte-Block zwischen diesen beiden Karten war also gar nicht zugewiesen. Da uns der Computer vom Eigentümer mit dem Hinweis »generell funktionsfähig« überlassen wurde, gingen wir zunächst davon aus, dass dieser auch korrekt konfiguriert sei. Denkbar ist ebenfalls, dass der Sol-20 zuvor einmal mit drei Karten (also mit 48 Kilobyte RAM) betrieben wurde und eine davon irgendwann aus dem Rechner entfernt wurde. Die Adressierung war zwar schon vorher auffällig, allerdings war zunächst nicht klar, dass diese »Speicherlücke« (englisch: memory gap) ein Problem

```

1 EN 0
2 0000: 22 88 01 21 00 00 22 8F 01 22 91 01 22 93 01 26
3 0010: 10 22 8D 01 97 32 95 01 01 94 01 01 91 01 87 0A
4 0020: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
5 0030: 00 3A 96 01 07 CD 42 01 F1 17 D2 24 8F 00 3A 96 01
6 0040: F5 CD 04 01 1E 03 CD 1F 01 1D C2 46 00 3A 96 01
7 0050: 07 CD 42 01 F1 17 DA 48 00 CD 04 01 3A 95 01 87
8 0060: 00 00 00 00 3A 96 01 01 32 00 00 00 00 00 C2 1A 00 00
9 0070: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
10 0080: CD 8A 00 2A 93 01 CD 8A 00 C3 15 84 01 82 00 2A 01
11 0090: 6F CD 83 2A 7C 17 67 CD 83 00 CD AE 00 15 C2 8E
12 00A0: 00 16 04 CD AE 00 1D C2 88 00 CD BF 00 09 3E 3E
13 00B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
14 00C0: 80 CD CA 00 3E 00 00 00 00 00 00 00 00 00 01 19 00 3A 88
15 00D0: 01 09 47 E3 C9 81 E3 00 C5 E5 01 17 00 2A 88
16 00E0: 09 E3 C9 FE C9 CD 7A 00 22 83 01 CD F7 00 2A
17 00F0: 88 23 23 23 E9 7E 23 F1 00 CA BF 00 00 CD CA
18 0100: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
19 0110: 95 01 F1 C9 F5 3A 00 00 00 00 00 00 00 00 00 00 00
20 0120: 29 81 CD A2 01 CD 04 01 C5 F3 CD 14 01 16 F1 18 F5
21 0130: 77 AE C4 5A 01 81 17 2C C2 2F 01 24 15 C2 2F 01
22 0140: F1 F5 CD 14 01 16 18 F3 AE C4 5A 01 11 17 2C
23 0150: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
24 0160: 95 01 87 07 87 07 85 0F 7E 00 88 77 2A 8F 01 23 22
25 0170: 8F 01 7C 85 CA E6 00 E1 C5 43 4F 40 59 4C 45 54
26 0180: 45 44 80 41 42 4F 52 54 45 44 00/
27

```

Abb. 10: Das Hex-Listing des Testprogramms, das in den Sol-20 eingeippt werden musste.

darstellen könnte.<sup>7</sup> Nachdem die zweite Karte so umadressiert worden war, dass sie direkt im Anschluss an Karte 1 liegt – die Karte besitzt dafür einen praktischen Jumperblock, sodass lediglich 4 Schalter umgestellt werden müssen, statt an der Karte zu löten – lief schließlich auch das Microsoft-BASIC.

Zur vollständigen Endkontrolle war in der Anleitung der Erweiterung ein Testprogramm als Hex-Listing angegeben (vgl. Abb. 10). Dieses besteht aus einer halben Seite Zahlenwerte, welche über das im ROM integrierte Monitorprogramm in den Rechner eingegeben werden können. Nach 30 Minuten Eingabezeit und mehreren Kontrollen lief dann auch dieses Programm und meldete keine Speicherfehler. Zur Verifikation wurde das Programm mit einem Kassettenrekorder auf einer handelsüblichen Compact Cassette abgespeichert und nach dem Einbauen eines Fehlers in der Speicherplatine, dem Entfernen eines Speicherchips, wieder in den Sol-20 eingelesen und erneut ausgeführt, wobei der fingierte Fehler korrekt gefunden wurde (vgl. Abb. 11 u. 12). Das Testprogramm war damit auch als korrekt funktionierend anzusehen und bestätigte, dass der Sol-20 nunmehr in vollem Umfang wieder funktional war.

## Zusammenfassung der Reparatur

Der Reparaturbericht beschreibt detailliert die Informationsquellen, Methoden, Werkzeuge und Materialien, die für die Reparatur des Computers im Rahmen hob-

7 In Computern späterer Zeit ist dieses Problem nicht mehr vorhanden. Bei genauerer Betrachtung scheint aber sinnvoll für »simple« Computer, dass RAM entweder an festen Adressen liegt oder, wenn, wie in diesem Fall, der Rechner ab Werk erweiterbar ist, in einem Block liegen muss, da dann die Adressen von der ersten RAM-Adresse durchgezählt werden können, bis kein RAM mehr vorhanden ist.



Abb. 11: Zu Testzwecken kann eine Karte von oben in den S-100-Bus gesteckt werden. Hier wird die scheinbar defekte RAM-Karte getestet.



Abb. 12: Zu Testzwecken wurde ein RAM-Baustein aus der Karte gezogen und ein Speichertest durchgeführt. Das »XG« im oberen linken Buchstabenblock zeigt den fehlenden Baustein.

byistischer Möglichkeiten zur Verfügung standen und angewandt wurden. Diese seien hier noch einmal stichpunktartig systematisierend zusammengefasst, um den Material-, Zeit- und ›Wissens-‹Aufwand zu verdeutlichen:

#### Werkzeuge:

- Multimeter zur Spannungs- und Widerstandsmessung
- Pinsel zur Staubbefreiung
- Step-down-Konverter zur Anpassung der Netzspannung

- Lötkolben und Heißklebepistole zur Herstellung des Tastatur-Testers
- Mobiltelefon als Audiowiedergabegerät und für Internetrecherchen
- 3,5-mm-Klinkenstecker-Kabel zum Anschluss des Computers an Smartphone und Kassettenrecorder
- Sol-20-Emulator zur Verifikation von Programmfunctionen
- Finger zum Andrücken der ICs, zur Herstellung von elektrischem Kontakt (Circuit Bending, vgl. Braguinski 2014), zum Wärmetest der DRAM-Bausteine und Programmeingabe
- Schraubenzieher zum Entfernen der Schaumstoffpad-Reste (der Sol-20 ist mit handlösbarer Schrauben verschlossen)
- Schere zum Ausschneiden der Kontaktfolie
- Audiokassettenrecorder und MP3-Recorder zum Speichern und Laden des abgetippten Programms

#### Arbeitsmittel:

- Isopropanol zur Reinigung von Kontakten und zur Entfernung von Klebstoffresten
- Mylarfolie, Aluminiumfolie und Metallschichtpapier als Ersatz für die Kontaktfolie
- Schaumstoffpads als Ersatz für die defekten originalen Schaumstoffpads
- Doppelseitiges Klebeband zur Befestigung der Kontaktfolie auf den Schaumstoffpads und an den Tasten
- Diverse elektronische Bauteile und Lochraster-Platine zum Aufbau des Tastatur-Testers
- Audiokassette
- Papier zum Ausdrucken des Hex-Codes des Speichertestprogramms

#### Diagnose-Methoden:

- Sichtprüfung auf mechanische Beschädigungen und Korrosionen
- Spannungsmessung der korrekten Spannungen am Netzeil und der internen Elektronik
- Widerstandsmessung zur Prüfung von Kontakten und zum Test der Metallfolie

#### Informationsquellen:

- Internet zum Ermittlung von Reparaturratschlägen, als Quelle für Schaltpläne und .wav-Dateien
- Reparatur-Erfahrung basierend auf vergleichbaren Problemen bei vorherigen Computern

#### Bezugsquellen:

- Eigene Material- und Bauteilsammlung: Isopropanol, elektronische Bauteile für Tastatur-Tester

- Elektronikbedarfshandel (Conrad Elektronik, Segor, Reichelt)
- Internet (eBay): Mylarfolie, Tastaturstempel

Markant ist insbesondere der Einsatz von Materialien und Werkzeugen, die im ›Hausgebrauch‹ (Isopropanol, Pinsel, ...) zur Verfügung stehen oder von Materialien, die für andere Zwecke bestimmt sind (Kassettenrecorder, Handy, Audiokassetten, Mylarfolie, Klebeband, ...), aber hier ›umgenutzt‹ werden. Ebenso ist die Verwendung der Finger als Werkzeuge zu Reparatur- und Testzwecken hervorzuheben; die Verwendung des Attributs ›hands-on‹ hat angesichts dieser Praktiken durchaus ihre Berechtigung.

Als Informationsquellen diente hier ausschließlich das Internet (und sekundär die von dort bezogenen Manuale, Schaltpläne und Reparaturtipps). Die dort gesammelten Informationen über den Sol-20 stammen sämtlich von Hobbyisten und sind dort mit dem Ziel hinterlegt worden, die Erinnerung wachzuhalten und die dazugehörigen Informationen über diesen seltenen Homecomputer zu archivieren. Anders als in einem Archiv oder Museum ist der Bestand dieser Informationen weder fachlich strukturiert noch katalogisiert oder gar gesichert; während der Reparaturarbeiten ›verschwand‹ die Seite mit den Sol-20-Programmen im .wav-Format. Das beständige Herunterladen, Verarbeiten, Erweitern und Wiederhochladen (›Spiegeln‹) von Informationen<sup>8</sup> gehört zum Modus Operandi der Retrocomputing-Szenen und sichert den Fortbestand, die Erweiterung und Weitergabe von Informationen.

## HISTORISCHE COMPUTER ALS EPISTEMISCHE DINGE

In der oben beschriebenen Reparatur zeigt sich bereits der eingangs genannte Hybridcharakter des zu reparierenden Computers als *Objekt und Werkzeug des Wissens*. Die Funktionalität des *Objektes* Sol-20 ließ sich nur erproben und bestätigen, indem man ihn darauf hin geprüft hat, ob er wieder als *Werkzeug* fungieren kann. Dies war aber nur möglich, indem man ihn, wie im abschließenden Experiment, mit einem Programm versehen hat, das seine Funktionalität bestätigt, indem der Computer ›sich selbst prüft‹. Im Folgenden soll dieser Doppelcharakter zu einer Epistemologie des Defekts umgedeutet werden, aus der sich – über zwei theoretische Ansätze – der ›Mehrwert‹ der *Reparatur als Wissensarbeit* begründen lässt. Diese epistemologische Perspektive stellt zudem die zentrale Motivation dar, aus der heraus ein Computerarchäologe ein medientechnisches Objekt repariert. Computerarchäologie geht es nicht darum, ein Objekt wieder in seinen ursprünglichen Funktionszusammenhang zu bringen, es wieder nutzbar zu machen, um so vielleicht konsumkritisch der Verwertungslogik von Industrieprodukten zu widersprechen.

8 Die .wav-Dateien mit den Sol-20-Programmen sind zwischenzeitlich wieder verfügbar gemacht worden (Nama o.J.).

Dies wäre angesichts der Leistungsmerkmale des Sol-20 (verglichen mit heutigen Computern) auch keine sinnvolle Verwendungweise. Es geht vielmehr darum, einen Computer zu einem Gegenstand des Wissens in medienwissenschaftlicher Perspektive zu machen. Ein historischer Computer kann auf diese Weise – über seine technomathematische und informative Beschreibung und über seine Reparatur – Element einer Auseinandersetzung mit der (Computer-)Geschichte werden.

Hierzu ist es zunächst notwendig, die Objektkategorien genauer zu definieren. Die Differenzierung von *Werkobjekt* und *Werkzeug* rekurriert auf Martin Heideggers Theorie der Zuhändigkeit. In *Sein und Zeit* (Heidegger 1967) definiert er zunächst die *Zuhändigkeit* (den Werkzeug-Charakter) am Beispiel eines Hammers:

»Zum Sein von Zeug gehört je immer ein Zeugganzes, darin es dieses Zeug sein kann, das es ist. Zeug ist wesenhaft ›etwas, um zu ...‹. Die verschiedenen Weisen des ›Um-zu‹ wie Dienlichkeit, Beiträglichkeit, Verwendbarkeit, Handlichkeit konstituieren eine Zeugganzheit. In der Struktur ›Um-zu‹ liegt eine Verweisung von etwas auf etwas. [...] In solchem gebrauchenden Umgang unterstellt sich das Besorgen dem für das jeweilige Zeug konstitutiven Um-zu; je weniger das Hammerding nur begafft wird, je zugreifender es gebraucht wird, um so ursprünglicher wird das Verhältnis zu ihm, um so unverhüllter begegnet es als das, was es ist, als Zeug. Das Hämmern selbst entdeckt die spezifische ›Handlichkeit‹ des Hammers. Die Seinsart von Zeug, in der es sich von ihm selbst her offenbart, nennen wir die Zuhändigkeit. [...] Das Eigentümliche des zunächst Zuhändenen ist es, in seiner Zuhändigkeit sich gleichsam zurückzuziehen, um gerade eigentlich zuhanden zu sein.« (Heidegger 1967: 68f.)

Mit anderen Worten: Werkzeug muss also, um als solches nutzbar zu sein, ›unsichtbar‹ bleiben und sich in seinem Gebrauch erschöpfen. Problematisch wird dies jedoch dann, wenn das Werkzeug ›in den Blick gerät‹ und damit wieder *vorhanden* ist, etwa, weil es defekt ist:

»Werkzeug stellt sich als beschädigt heraus, das Material als ungeeignet. Zeug ist hierbei in jedem Falle zuhanden. Was aber die Unverwendbarkeit entdeckt, ist nicht das hinsehende Feststellen von Eigenschaften, sondern die Umsicht des gebrauchenden Umgangs. In solchem Entdecken der Unverwendbarkeit fällt das Zeug auf. Das Auffallen gibt das zuhandene Zeug in einer gewissen Unzuhandenheit. Darin liegt aber: das Unbrauchbare liegt nur da –, es zeigt sich als Zeugding, das so und so aussieht und in seiner Zuhändigkeit als so aussehendes ständig auch vorhanden war. Die pure Vorhandenheit meldet sich am Zeug, um sich jedoch wieder in die Zuhändigkeit des Besorgten, d.h. des in der Wiederinstandsetzung Befindlichen, zurückzuziehen. Diese Vorhandenheit des Unbrauchbaren entbehrt noch nicht schlechthin jeder Zuhändigkeit, das so vorhandene Zeug ist noch nicht ein nur irgendwo vorkommendes Ding. Die Beschädigung des Zeugs ist noch nicht eine bloße Dingveränderung, ein lediglich vorkommender Wechsel von Eigenschaften an einem Vorhandenen. [...] Die Struktur des Seins von Zuhändenem als Zeug ist durch die Verweisungen bestimmt. [...] Die Verweisungen selbst sind nicht betrach-

tet, sondern ›da‹ in dem besorgenden Sichstellen unter sie. In einer Störung der Verweisung – in der Unverwendbarkeit für [...] wird aber die Verweisung ausdrücklich.« (Ebd.: 73f.)

Die Störung macht das Werkzeug wieder sichtbar. Das defekte Werkzeug jedoch zeigt eine spezifische Eigenart in seiner (durch den Effekt verursachten) Vorhandenheit, die es von anderen vorhandenen Dingen (etwa den Werkstücken) unterscheidet: Es provoziert dazu, seine Zuhandenheit zu imaginieren und bestenfalls zu restituieren (es zu reparieren). Heidegger entwickelt anhand dieses Doppelcharakters eine Unterscheidung von Praxis und Theorie:

›Der nur ›theoretisch‹ hinsehende Blick auf Dinge entbehrt des Verstehens von Zuhandenheit. Der gebrauchend-hantierende Umgang ist aber nicht blind, er hat seine eigene Sichtart, die das Hantieren führt und ihm seine spezifische Dinghaftigkeit verleiht. [...] Das ›praktische‹ Verhalten ist nicht ›atheoretisch‹ im Sinne der Sichtlosigkeit, und sein Unterschied gegen das theoretische Verhalten liegt nicht nur darin, daß hier betrachtet und dort gehandelt wird, und daß das Handeln, um nicht blind zu bleiben, theoretisches Erkennen anwendet, sondern das Betrachten ist so ursprünglich ein Besorgen, wie das Handeln seine Sicht hat. Das theoretische Verhalten ist unumsichtiges Nur-hinsehen. Das Hinsehen ist, weil unumsichtig, nicht regellos, seinen Kanon bildet es sich in der Methode.« (Ebd.: 69)

Beides, Theorien über die Dinge und Praktiken ihrer Verwendung, vereint sich in Heideggers Hammer-Beispiel. Diese Differenzierung und gleichzeitige Verschmelzung ist wissenschaftstheoretisch später durch Hans-Jörg Rheinberger wieder aufgegriffen und auf die Vorgehensweise wissenschaftlicher Untersuchungen, Untersuchungsobjekte und Untersuchungsapparate übertragen worden.

Rheinberger unterscheidet zwischen »epistemischen Dingen« (»Forschungsgegenstand, Wissenschaftsobjekt«) und »technischen Dinge[n]« (»Experimentalbedingungen«) in Experimentalsystemen:

»Im Gegensatz zu den epistemischen Objekten müssen die Experimentalbedingungen innerhalb der jeweils gültigen Reinheits- und Präzisionsstandards von charakteristischer Bestimmtheit sein. Die Wissenschaftsobjekte werden von den Experimentalbedingungen im doppelten Sinne ›eingefäßt‹: Sie werden eingebettet und durch die Umfassung gleichzeitig begrenzt.« (Rheinberger 2000: 53)

Epistemische Dinge können zu technischen Dingen werden durch Black Boxing:

»Allerdings reflektiert dieser Ausdruck nur die eine Seite des Übergangs von einem epistemischen zu einem technischen Ding: den nachmaligen Routine-Charakter des transformierten Objekts. Mindestens ebenso wichtig sind jedoch die Auswirkungen dieses Vorgangs auf die neue Generation epistemischer Dinge, die gerade im Entstehen begriffen sind, die Eröffnung neuer Möglichkeiten der Untersuchung.« (Rheinberger 2001: 26)

Technische Dinge können andererseits allerdings auch zu epistemischen Dingen werden, wenn sie Fragen aufzuwerfen beginnen:

»Technische Gegenstände haben die Zwecke zu erfüllen, für die sie gebaut worden sind, es sind Maschinen, die definierte Antworten geben. Ein epistemisches Objekt hingegen ist in erster Linie eine Maschine, die Fragen aufwirft. Es ist in und aus sich selbst heraus nicht ein technisches Ding.« (Rheinberger 2000: 55)

Eine der Möglichkeiten, wie ein technisches Ding Fragen aufwerfen kann, ist, wenn es *defekt* wird. Ein defekter Computer, sei er nun total ausgefallen oder in bestimmten Funktionen gestört, führt dem Nutzer, der ihn versucht als Werkzeug zu verwenden, permanent seine Vorhandenheit vor Augen. Anstelle der Nutzungsfragen entstehen Fragen nach der Ursache des Defekts und seiner spezifischen Verfassung, womit die technischen Details der Maschine in den Fokus geraten. Das epistemische Objekt »defekter Computer« stellt die Frage nach seiner Verfassung ins Zentrum.

Mit diesen Darstellungen steht nun eine Theorie zur Verfügung, die als Epistemologie des Reparierens genügen kann, wenn *die Tätigkeit des Reparierens selbst* noch in die Betrachtung einbezogen wird. Hier schlägt Rheinberger den Begriff des Bastelns vor, weil der Bastler experimentierend und nicht nach Schema vorgeht und weil Experimentatoren »ihre Geschichte nicht im Voraus erzählen [können]«, d.h. stetig Neues und Unbekanntes hervorbringen:

»Ein Wissenschaftler ist vor allem ein ›Bastler‹, ein ›bricoleur‹, und nicht ein Ingenieur. In ihrem nicht-technischen Charakter transzendierte das experimentelle Ensemble die technischen Objekte, aus denen es zusammengesetzt ist.« (Ebd.: 55f.)

Sein Blick auf das Experimentsystem fasst also stets den Doppelcharakter von technischen Objekten ins Auge. Die Verwendung des Werkzeugs geschieht für ihn deshalb auch stets reflektiert. Die Klassifikation des Experimentierens als »Basteln« und damit des Wissenschaftlers als Bastler besitzt für Rheinberger aber zusätzlich noch eine wissenschaftshistorische Relevanz, führt sie doch die *Kontingenz von Entdeckungen* zurück in den Prozess der Wissenschaft:

»Dieses wilde Denken innerhalb der Wissenschaft, ohne das Rationalität, Logik und Präzision stumm und unfruchtbare blieben, diese ›nächtliche Wissenschaft‹, diese ›Werkstatt des Möglichen‹, dieses Labyrinth der Irrungen und Wirrungen ist gegen eine Geschichtsschreibung zu verteidigen, die versucht, die Entwicklung der modernen Wissenschaften in Standardisierung, Normierung und Regulierung aufgehen zu lassen. Auch die Präzisionsmessung gehört in diesen Zusammenhang, sofern sie nicht selbst als Forschungsprozeß betrieben wird. Dagegen setze ich den Begriff des Bastelns und bestehe auf den Werten des Unpräzisen, des Vor-Normativen, des nicht definierbaren und nicht standardisierbaren Überschusses als unverzichtbare Elemente des Forschungsprozesses, jenes ganzen epistemischen Konti-

nents, der zu Beginn dieses Jahrhunderts als Kontext der Entdeckung so erfolgreich exterritorialisiert und den Psychologen überlassen wurde.« (Ebd.: 56)

In der Folge ließe sich der oben dargestellte Reparaturprozess, vorgenommen von einem Bastler an einem ihm zuvor unbekannten Computer in diese Beschreibung integrieren. Wenn im Defekten aber Vorhandenes und Zuhändenes einander überlappen, technische Dinge zu epistemischen Dingen werden und die Reparaturmethoden sich zugleich an heuristischen Verfahren (*trial and error*) und autodidaktisch erworbenen, standardisierten Methoden orientieren, dann restituiert die Reparatur nicht bloß das vormalige Werkzeug als technisches Objekt, sondern stellt zugleich eine Form von *Wissensarbeit* dar. Diese wird, angewendet auf ein historisches Objekt, welches aus dem dysfunktionalen Zustand von ›Hardware‹ in den operativen Zustand von ›Computer‹ versetzt wird, zugleich ein Akt *operativer Historiographie*. Die Demonstration des funktionierenden Computers tritt neben den technischen Werkstattbericht und dessen theoretischer Flankierung als eine Einheit – als *Geschichtsarbeit*.

Der funktionierende Sol-20 erhält seinen Werkzeug-Charakter zurück, wird aber zugleich, weil er technisch obsolet ist, nie mehr als reines Werkzeug genutzt werden. Er bleibt selbst dann, wenn er hinter seiner Funktionalität verschwindet – etwa, indem man ein Computerspiel auf ihm spielt – stets ein Beispiel sowohl für operative Geschichtsarbeit als auch für den Blick in die Black Box, die (funktionierende) Computer als Werkzeuge notwendig sein müssen. Für John von Neumann (und alle ihm nachfolgenden Computerarchitekten) war/ist der Computer noch ein epistemisches Ding, dass es im Zuge des Black Boxing zu einem technischen Ding zu machen gilt. Für den Bastler stellt die »Black Box Computer« (Becker 2012) wieder die reine Herausforderung eines epistemischen Dings dar.

## SCHLUSS

Abschließend möchten wir die Frage aufwerfen, bis zu welchem Beschädigungsgrad ein Computer überhaupt noch repariert werden kann – oder mit anderen Worten: ab wann er seinen Status als epistemisches Ding wieder verliert. Im Unterschied zu anderen technischen Artefakten sind Computer Apparate, deren Werkzeug-Charakter darin besteht, gerade nicht das zu sein, was sie sind, weil ihre Funktion in der Simulation anderer Maschinen/Medien besteht. Sie sind auf der höchsten Abstraktionsstufe Architekturen, die sich mit Funktionsdiagrammen beschreiben lassen. Der eingangs in Erinnerung gerufene »First Draft« John von Neumanns leistet in seinem Black Boxing genau dies: aus der materiellen Maschine eine diagrammatische zu machen, die über ihre prinzipiellen Funktionen beschrieben werden kann.

Diese Beschreibbarkeit hat Konsequenzen für die Frage nach dem maximalen Beschädigungsgrad: Selbst dann, wenn der zu ›reparierende‹ Computer gar nicht

mehr existiert<sup>9</sup> oder sogar nie existiert hat,<sup>10</sup> lässt er sich als materielles, operatives Objekt restituiieren, um im Anschluss wieder voll funktionsfähig zu sein. Diese restituierte Funktionalität kann jedoch nicht mehr in seiner Materialität liegen, sondern darin, dass der Charakter »des Computers« darin besteht, andere Maschinen zu sein. Sobald er dieses Vermögen (wieder) besitzt, wird man ihn als repariert ansehen dürfen. Dass man aber auch hierfür die Ebene der abstrakten Diagramme verlassen muss, um sich der konkreten Technik zuzuwenden, schließt den in diesem Beitrag gezeichneten diskursiven Kreis.

## LITERATUR

- Anonymous (o.J.): [https://public.me.com/lord\\_philip](https://public.me.com/lord_philip) (nicht mehr online, zuletzt abgerufen 01.07.2016).
- Battle, Jim (2005): »Solace – Sol Anachronistic Computer Emulation«, <http://www.sol20.org/solace.html> (zuletzt abgerufen 27.07.2017).
- Battle, Jim (2006): Sol-20.org, <http://www.sol20.org/> (zuletzt abgerufen 25.11.2016).
- Becker, Rainer C. (2012): Black Box Computer. Zur Wissensgeschichte einer universellen kybernetischen Maschine, Bielefeld.
- Braguinski, Nikita (2014): »Circuit Bending. Ein unheimlicher Spaß«, in: Retro 31, S. 38-39.
- Burnet, M. M./Supnik, R. M. (1996): »Preserving Computing's Past. Restoration and Simulation«, in: Digital Technical Journal 8 (3), S. 23-38.
- Ernst, Wolfgang (2004): »Das Gesetz des Sagbaren. Foucault und die Medien«, in: Peter Gente (Hg.), Foucault und die Künste, Frankfurt/M., S. 238-259.
- Ernst, Wolfgang (2012): Gleichursprünglichkeit. Zeitwesen und Zeitgegebenheit technischer Medien, Berlin.
- Foucault, Michel (2003): »Das Spiel des Michel Foucault«, in: ders., Schriften in vier Bänden. Dits et écrits, Bd. 3: 1976-1979, S. 391-429.
- Heidegger, Martin (1967 [1927]): Sein und Zeit, 11., unveränd. Aufl., Tübingen.
- Höltgen, Stefan (2014): »All Watched Over by Machines of Loving Grace« – Öffentliche Erinnerung, demokratische Informationen und restriktive Technologien am Beispiel der »Community Memory«, in: Ramón Reichert (Hg.), Big Data.

- 
- 9 Ein Beispiel hierfür ist das Restaurationsprojekt von Horst Zuse, der für das Deutsche Technikmuseum in Berlin den Z3-Computer seines Vaters nachgebaut hat, von dem kein vollständiges, geschweige denn funktionierendes Exemplar mehr existiert – wohl aber die technischen Beschreibungen (Zuse 2011).
- 10 Als Beispiel hierfür mag die ebenfalls im Berliner Technikmuseum erstellte »Machina arithmeticæ dyadicæ« von G.W. Leibniz dienen, die dieser 1679 auf dem Papier entworfen, aber nie gebaut hatte. Die Konstruktion fand 1968/1972 in München statt (Leibniz 1679; Stein 2016).

- Analysen zum digitalen Wandel von Wissen, Macht und Ökonomie, Bielefeld, S. 385-404.
- Hopper, Grace (1947): »Log Book With Computer Bug«, in: Grace Murray Hopper Collection, Archives Center, National Museum of American History, Smithsonian Institution, [http://americanhistory.si.edu/collections/search/object/nmah\\_334663](http://americanhistory.si.edu/collections/search/object/nmah_334663) (zuletzt abgerufen 27.07.2017).
- IEEE 696-1983, online: [http://www.imsai.net/download/IEEE\\_696\\_1983.pdf](http://www.imsai.net/download/IEEE_696_1983.pdf) (zuletzt abgerufen 27.07.2017).
- Johnson, Herb (2012): »Processor Tech Sol – Keyboard Repair«, [http://www.retro-technology.com/restore/sol\\_keys.html#display](http://www.retro-technology.com/restore/sol_keys.html#display) (zuletzt abgerufen 27.07.2017).
- Leibniz, Gottfried Wilhelm (1679): »De progressione dyadica«, digitalisierte Manuskriptseite, in: LeibnizCentral – Virtuelle Ausstellung, <http://dokumente.leibnizcentral.de/index.php?id=95> (zuletzt abgerufen 25.11.2016).
- Levy, Steven (2010): Hackers. Heroes of the Computer Revolution, Peking u.a.
- Luhmann, Niklas (1977): »Differentiation of Society«, in: The Canadian Journal of Sociology / Cahiers Canadiens de Sociologie 2 (1), S. 29-53.
- Montfort, Nick (2013): »Beyond the Journal and the Blog. The Technical Report for Communication in the Humanities«, in: Amodern 1: The Future of the Scholarly Journal, <http://amodern.net/article/beyond-the-journal-and-the-blog-the-technical-report-for-communication-in-the-humanities/> (zuletzt abgerufen 25.11.2016).
- Nama, Philip (o.J.): »Neoncluster. Projects and Articles: Processor Technology Sol-20 .wav Files«, <http://www.neoncluster.com/projects-sol20/sol20-wav.html> (nicht mehr online, zuletzt abgerufen 25.11.2016).
- Rheinberger, Hans-Jörg (2000): »Experiment: Präzision und Bastelei«, in: Christoph Meinel (Hg.), Instrument/Experiment. Historische Studien, Berlin/Diepholz, S. 52-60.
- Rheinberger, Hans-Jörg (2001): Experimentalsysteme und epistemische Dinge. Eine Geschichte der Proteinsynthese im Reagenzglas, Göttingen.
- Roberts, Henry Edward/Yates, William (1975): »Altair 8800. The Most Powerful Minicomputer Project Ever Presented – Can be Built for Under \$400«, in: Popular Electronics 7 (1), S. 33-38.
- Stein, Erwin (2016): »Leibniz-Ausstellung der Leibniz-Universität Hannover«, <https://www.uni-hannover.de/de/universitaet/leibniz/leibnizausstellung/> (zuletzt abgerufen 25.11.2016).
- Takhteyev, Yuri/DuPont, Quinn (2013): »Retrocomputing as Preservation and Remix«, in: iConference 2013 Proceedings, S. 422-432, <http://hdl.handle.net/2142/38392> (zuletzt abgerufen 27.07.2017).
- Tanenbaum, Andrew S. (2005): Structured Computer Organization, Upper Saddle River, NJ.
- von Neumann, John (1993): »First Draft of a Report on the EDVAC«, in: EEE Annals of the History of Computing 15 (4), S. 27-76.

- Weiser, Mark (1991): »The Computer for the 21<sup>st</sup> Century«, in: Scientific American 265 (3), S. 94-104.
- White, Hayden (1994): »Der historische Text als literarisches Kunstwerk«, in: Christoph Conrad/Martina Kessel (Hg.), Geschichte schreiben in der Postmoderne, Stuttgart, S. 123-157.
- Zaks, Rodnay (1981): Don't! Or How to Care for Your Computer, Berkeley u.a.
- Zuse, Horst (2011): »70 Jahre Rechner Zuse Z3. Nachbau der Z3 durch Horst Zuse«, <http://www.horst-zuse.homepage.t-online.de/horst-zuse-z3-html/z3-broschuere-mit%20foto-allgemein-2011-1.pdf> (zuletzt abgerufen 25.11.2016).

## ABBILDUNGSVERZEICHNIS

Abb. 1: Cover Popular Electronics 10 (1) (1976), [http://www.swtpc.com/mholley/PopularElectronics/Jul1976/PE\\_Jul\\_1976\\_Cover.jpg](http://www.swtpc.com/mholley/PopularElectronics/Jul1976/PE_Jul_1976_Cover.jpg) (zuletzt abgerufen 21.07.2017).

Abb. 2-12: Photographien, Stefan Höltgen und Marius Groth 2016.