RHEINISCHE FRIEDRICH-WILHELMS-UNIVERSITÄT BONN

INSTITUT FÜR INFORMATIK IV

BACHELORARBEIT

Aufnahme und Wiedergabe von Tastatur-Eingabesequenzen mittels Arduino Mikrocontroller

ANDREAS J. FRITZ, 2404696

ERSTGUTACHTER: PROF. DR. MICHAEL MEIER

ZWEITGUTACHTER: DR. MATTHIAS FRANK

BONN, 23 SEPTEMBER 2014

Inhaltsverzeichnis

Abbildungsverzeichnis

Kapitel 1

Einleitung

Die Verwendung von Geräten zum Erfassen von Tastatur-Eingabesequenzen, sogenannten Keyloggern, ist schon seit Mitte der 1970er Jahre publik [Eng87]. Die New York Times berichtete zu dieser Zeit von einer Spionage durch solche Geräte in US-Botschaften und -Konsulaten in Moskau und St. Petersburg, bei welcher IBM Selectric typewriter angegriffen wurden. Es existieren derzeit sowohl Software- als auch Hardware-Keylogger, jedoch lassen sich diese auch noch einmal in verschiedene Unterkategorien unterteilen. So ist z.B. ein Adapter, welcher zwischen Tastatur und PC steckt, wie in Abbildung 1.1 [keya] oder 1.2 [keyb] dargestellt, eine mögliche Implementierung eines Hardware-Keyloggers. Diese existieren sowohl für PS/2- als auch USB-Tastaturen und sind im Handel frei erhältlich [kee].

Allerdings gelten auch andere Geräte als Hardware-Keylogger, wie z.B. Keypads, welche bei Geldautomaten über das PIN-Feld gelegt werden um den PIN-Code zu erfassen [Kir08]. Über Schäden verursacht durch Keylogger existieren allerdings nur wenige Informationen, da Straftaten im Bereich Computerbetrug und Spionage oftmals nicht erkannt oder nicht gemeldet werden [Bun12]. So können nur beispielhaft monetäre Erwartungswerte über Schwarzmärkte und spezielle Be-

Abbildung 1.1: Keylogger PS/2

Abbildung 1.2: Keylogger USB

trugsdelikte, wie z.B. Kreditkartenbetrug gebildet werden [TH08], oder es werden einzelne Straftaten publik, wie u.a. der versuchte Raub von $423 Millionen in London [Kei05].

Jedoch ist nicht nur das Mitlesen von Tasteneingaben über die Tastaturschnittstelle möglich, sondern auch die Wiedergabe von Tastatur-Eingabesequenzen, wie u.a. auf der Blackhat Conference demonstriert wurde [Che09]. Hierbei wurde die Firmware des Mikrocontrollers derart überschrieben, dass sie nach einer Tasteneingabe und einem zusätzlichen Befehl die Eingabe nochmals in umgekehrter Reihenfolge widergab.

Die vorliegende Bachelorarbeit mit dem Titel "Aufnahme und Wiedergabe von Tastatur-Eingabesequenzen mittels Arduino Mikrocontroller" soll jeweils durch eine Implementierung zeigen, dass es einerseits möglich ist Signale einer PS/2-Tastatur mithilfe des Arduino Mikrocontroller [arda] abfangen und speichern zu können. Andererseits soll gezeigt werden, dass es möglich ist Tastatursignale durch den Mikrocontroller an ein Betriebssystem senden zu können.

Im Folgenden wird sowohl die Idee hinter diesem Thema, als auch mögliche Anwendungen zur Motivation näher beschrieben. Anschließend wird die geplante Herangehensweise für die Bearbeitung dieser Aufgabe geschildert.

1.1 Motivation

Das Aufnehmen und Wiedergeben von Tastatur-Eingabesequenzen bietet viele Möglichkeiten zur Implementierung von nützlichen Funktionalitäten. Im Rahmen dieser Bachelorarbeit dienen drei dieser Funktionalitäten als Motivation und werden dementsprechend mithilfe des Arduino Mikrocontroller [arda] implementiert:

Die erste Funktionalität ist das einfache Aufzeichnen und Abspeichern der Tastatur-Eingabesequenzen. Dabei sollen die Aufzeichnungen auf einer SD-Karte gespeichert werden, welche beliebig ausgetauscht werden kann.

Als zweite Funktionalität ist das Senden von Tastatursignalen an das Betriebssystem gedacht, welche als Skript auf einer SD-Karte hinterlegt sein können. Nach dem Aufrufen einer Konsole mittels Tastaturkürzeln, die abhängig vom jeweiligen Betriebssystem sind, kann so jeglicher Befehl auf dem System ausgeführt werden. Durch den Einsatz der SD-Karte sind die Skripte austauschbar, sodass verschiedenste Anwendungsmöglichkeiten bestehen.

Die dritte Funktionalität beinhaltet auch das Senden von Tastatursignalen an das Betriebssystem, jedoch werden diese über Ethernet an den Mikrocontroller übertragen. Dies ermöglicht, sofern der Zugriff auf eine Konsole möglich ist, die

Steuerung eines Betriebssystems in Echtzeit. Damit gleicht diese Funktionalität einem Remote-Zugriff, jedoch ohne den Einsatz von Software auf dem zu steuernden Betriebssystem.

Da es sich bei den letzten beiden Funktionalitäten um das Senden von Tastatursignalen über das PS/2-Protokoll handelt, besteht weiterhin die Möglichkeit, dass die eingegebenen Befehle ohne eine explizite Prüfung des Betriebssystems oder eines Virenscanners ausgeführt werden können. Dies zu evaluieren ist somit ein weiterer Bestandteil dieser Bachelorarbeit und kann mit Folgen für die IT-Sicherheit verbunden sein.

1.2 Aufbau der Arbeit

Zu Beginn dieser Bachelorarbeit ist im ersten Kapitel die Recherche bezüglich der PS/2-Tastaturschnittstelle und des PS/2-Protokolls für das weitere Vorgehen erforderlich . Zudem werden verwandte Arbeiten aus dem Bereich der Aufnahme und Wiedergabe von Tastatureingabesquenzen beschrieben und die rechtlichen Grundlagen benannt. Im nächsten Kapitel wird die Implementierung der Funktionalitäten dokumentiert. Dies deckt sowohl die notwendigen Softwarekomponenten für den Arduino Mikrocontroller ab, als auch den Aufbau der Elektronik. In dem darauf folgenden Kapitel werden die Ergebnisse der Implementierung evaluiert und bestehende Abwehrmechanismen beschrieben. Abschließend wird in dem letzten Kapitel die Bachelorarbeit zusammengefasst und ein Ausblick für zukünftige Arbeiten gegeben.

Kapitel 2

Grundlagen

Um den Mikrocontroller zu implementieren, der Tastatureingabesequenzen sowohl aufnehmen als auch wiedergeben soll, müssen dafür einige Grundlagen benannt werden. Dieses Kapitel beschreibt im Folgenden die benötigten Elemente der PS/2-Tastaturschnittstelle und des PS/2-Protokolls, sowie einige verwandte Arbeiten in diesem Themengebiet als auch die rechtlichen Grundlagen. Die Abschnitte PS/2-Tastaturschnittstelle und PS/2-Protokoll fassen die Beschreibungen von Adam Chapweske [Cha03] bzw. der Übersetzung von Bernward Mock [Moc05] zusammen.

2.1 PS/2-Tastaturschnittstelle

IBM entwickelte 1987 die PS/2-Tastatur zur Verwendung am gleichnamigen PC, dem Personal System/2, und ist kompatibel mit der zuvor entwickelten AT-Tastatur. Der Anschluss erfolgt über einen 5- oder 6-poligen Mini-DIN Stecker, bzw. alternativ über einen SDL-Stecker. In Abbildung 2.1 [fem] ist die Anordnung der 6 Pins aufseiten des PCs (female) zu sehen. Spiegelverkehrt dazu ist der Anschluss der Tastatur (male).

Für die Verwendung einer PS/2-Tastatur werden nur 4 der 6 Pins benötigt, da ein Datensignal, eine Erdung, ein Takt und eine Leitung mit 5 Volt ausreichen um Tastensignale zu übertragen (siehe Abbildung 2.2). Ein in der Tastatur verbauter Mikrocontroller, ein sogenannter Keyboard-Encoder, scannt die Tasten und überprüft ob eine Taste gedrückt ist oder nicht. PS/2-Tastaturen verwenden typischerweise zwischen 84 bis 104 Tasten, welche sogenannten Scancodes zugeordnet werden. Es existieren drei Scancode-Sets, wobei PS/2-Tastaturen den

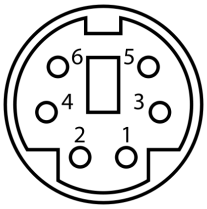

Abbildung 2.1: PS/2 Female Pins

Pin 1	Daten
Pin 2	kein Signal
Pin 3	Erdung
Pin 4	5 Volt
Pin 5	Takt
Pin 6	kein Signal

Abbildung 2.2: Pin Spezifikation

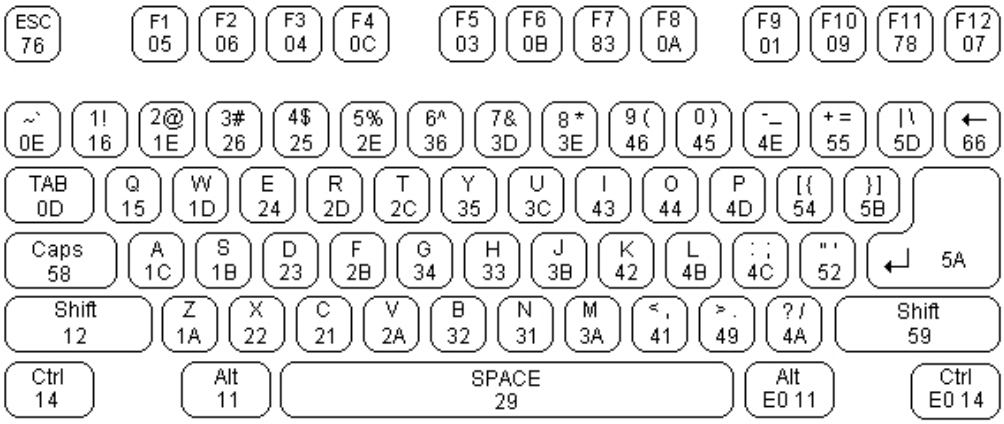

Abbildung 2.3: Scancode-Set 2 Ausschnitt

als Scancode-Set 2 bekannten Satz benutzen. In Abbildung 2.3 [sca] ist ein Ausschnitt dieser Scancodes auf einigen Tasten zu sehen und im Anhang befindet sich die Tabelle A.1 des gesamten Scancode-Sets 2.

Die Scancodes sind Hexadezimalwerte bestehend aus einem Makecode, welcher gesendet wird wenn die Taste gedrückt wird, und einem Breakcode, der beim Loslassen der Taste gesendet wird. Breakcodes setzen sich in fast allen Fällen aus einem 0xf0 und dem Makecode der Taste zusammen. Zudem existieren erweiterete Tasten, deren Makecode länger als ein Byte ist und zusätzlich ein 0xe0 als erstes Byte haben, was auch für den zugehörigen Breakcode gilt. Um also z.B. ein “G” wiederzugeben ist es notwendig zuerst die Shift-Taste gedrückt zu halten, die G-Taste zu drücken und beide Tasten in umgekehrter Reihenfolge loszulassen. Dementsprechend können für dieses Beispiel die folgenden Scancodes übertragen werden: 0x12 (Make L Shift), 0x34 (Make G), 0xf0 0x34 (Break G) und 0xf0 0x12 (Break L Shift).

Wenn eine Taste dauerhaft gedrückt wird setzt die Wiederholfunktion des Mikrocontrollers der Tastatur ein, auch Typematic genannt. Diese sendet mit einer gewissen Verzögerung (typematic delay) den Makecode der zuletzt gedrückten Taste und dann mit einer bestimmten Wiederholrate (typematic rate) fortwährend denselben Makecode, bis die Taste losgelassen wird. Beide Parameter können durch den PC, in diesem Zusammenhang auch Host genannt, eingestellt werden, wobei die Verzögerung zwischen 0,25 und 1,00 Sekunden liegen kann und die Wiederholrate zwischen 2,0 cps und 30,0 cps (Zeichen pro Sekunde).

Die Tastatur kann weiterhin einen Reset vollziehen und führt dabei einen Selbsttest, auch Basic Assurance Test (BAT) genannt, durch. Dabei wird die Verzögerung auf 0,5 Sekunden und die Wiederholrate auf 10,9 cps gesetzt, sowie Scancode-Set 2 geladen. Zudem werden zu Beginn des BAT die drei LEDs der Tastatur an und danach wieder ausgeschaltet, sowie 0xaa an den Host gesendet für ein erfolgreich abgeschlossenen BAT. Dies geschieht u.a. bei einem erstmaligem Anschluss der Tastatur an den Host.

Die Kommunikation zwischen dem Host und der Tastatur wird im folgenden Abschnitt anhand des PS/2-Protokolls beschrieben.

2.2 PS/2-Protokoll

Bei dem PS/2-Protokoll handelt es sich um ein sogenanntes bi-direktionales serielles Protokoll. Dies bedeutet, dass auch der Host Befehle an die Tastatur senden kann, im Fall des PS/2-Protokolls sind es 17 Host-Befehle. Dabei sind die folgenden Besonderheiten zu beachten:

- Die Tastatur löscht ihren Ausgabepuffer bei jedem empfangenen Befehl.
- Wenn die Tastatur einen ungültigen Befehl oder Parameter empfängt, muss sie mit 0xfe (Resend) antworten.
- Die Tastatur darf keine Scancodes senden, während sie einen Befehl verarbeitet.
- Wenn die Tastatur auf einen Parameter wartet, jedoch einen Befehl empfängt, wird der letzte Befehl verworfen.

Eine detaillierte Auflistung dieser Befehle zeigt die Tabelle ... im Anhang. Zu diesen Befehlen zählen 0xff (Reset), worauf die Tastatur 0xfa (Acknowledge) sendet

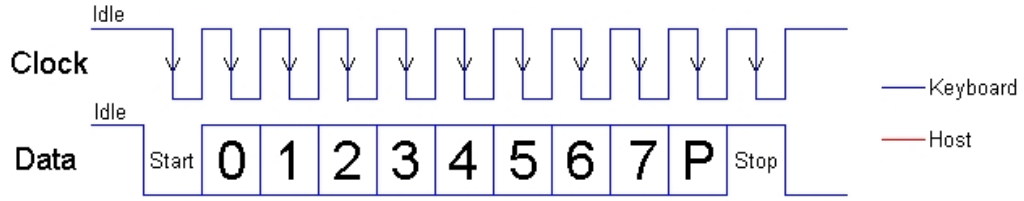

Abbildung 2.4: Kommunikation von Tastatur zu Host

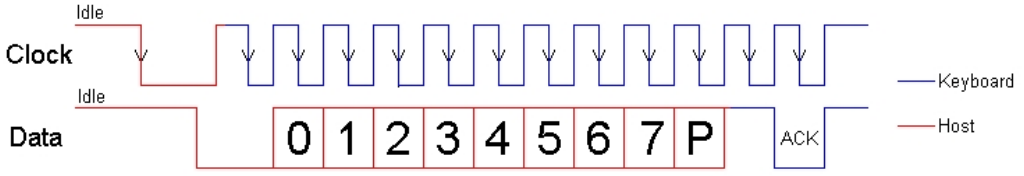

Abbildung 2.5: Kommunikation von Host zu Tastatur

und einen Reset durchführt, 0xfe (Resend), welcher die Tastatur das letzte gesendete Byte erneut senden lässt, und mehrere Set-Befehle wie z.B. * 0xf0 (Set Scancode-Set), der mit einem Parameter für * das Scancode-Set verändern kann.

Die Kommunikation von der Tastatur zum Host erfolgt in einem festgelegten Ablauf, welcher in Abbildung 2.4 dargestellt ist. Zu Beginn werden die Daten- und die Taktleitung auf High bzw. 1 gesetzt, welches der Ruhezustand ist. Danach wird 1 Startbit bei fallender Taktflanke gesendet, welches immer Low bzw. 0 ist. Im selben Rhytmus werden dann 8 Datenbits gesendet und zwar mit Least Significant Bit (LSB) voran. Anschließend folgt 1 Paritätsbit, High bzw. 1 bei ungerader Parität, und 1 Stopbit, welches immer High bzw. 1 ist. Mithilfe dieses Ablaufs wurde 1 Byte von der Tastatur zum Host übertragen.

Der Ablauf der Kommunikation von dem Host zur Tastatur folgt einem ähnlichen Schema, welches in Abbildung 2.5 zu sehen ist. ggf. 1 ACK-Bit (nur bei Host zu Tastatur) Delay

2.3 Verwandte Arbeiten

Wie bereits in der Einleitung erwähnt, existieren viele Produkte im Bereich der Hardware-Keylogger. So gibt es bereits Keylogger für USB-Tastaturen und PS/2-

Tastaturen mit verschiedenen Speichergrößen oder der Möglichkeit die aufgezeichneten Tastatureingaben über Wi-Fi zu versenden [kee].

Im Bereich der Mikrocontroller gibt es verschiedene Bibliotheken, die das Mitlesen von Tastatureingaben ermöglichen. Eine verbreitete Implementierung ist eine Bibliothek, die sowohl für Arduino Mikrocontroller als auch andere Mikrocontroller gedacht ist [ps2c]. Die bereitgestellten Funktionen erlauben es, wie in den mitgelieferten Beispielen gezeigt wird, die Tastatureingaben, der mit dem Mikrocontroller verbundenen Tastatur, über den Mikrocontroller auszugeben. Auch in anderen Implementierungen, wie z.B. dem Tastaturtreiber des Betriebssystems PrettyOS, werden Tastatureingaben entgegen genommen und in ASCII-Zeichen umgewandelt, um diese u.a. auf dem Bildschirm auszugeben [pre].

Es wurden aber auch Konzepte und deren Umsetzung dokumentiert, welche die Manipulation von Tastatureingaben zeigen. In einem bestehenden Ansatz wurde die Firmware des Mikrocontrollers einer Apple-Tastatur überschrieben [Che09]. Dies hatte zur Folge, dass nach einer normalen Zeicheneingabe und einer bestimmten Befehlsequenz diese Zeicheneingabe erneut, aber spiegelverkehrt an den PC gesendet wurde.

Andere Ansätze werden zudem als Produkt vertrieben, wie z.B. der USB-Stick Rubber Ducky [duc]. Dieser enthält unter seiner Abdeckung einen zusätzlichen Mikrocontroller mit einer Speicherkarte. Mithilfe von eigenen Befehlen, die in einer Textdatei auf der Speicherkarte gespeichert werden kann, führt der USB-Stick diese Befehle als Tastatureingaben aus, sobald er mit einem PC verbunden wird.

Zuletzt wurde ein weiterer Ansatz mit dem Namen BadUSB präsentiert [KN14]. Dabei wurde die Möglichkeit genutzt die Firmware eines jeglichen USB-Geräts umzuschreiben, sodass diese sich als eine andere Geräteklasse ausgeben, z.B. Tastatur. Diese wiederum sendet für den Anwender nicht wahrnehmbare schnelle Tasteneingaben an den PC, sodass dieser Malware aus dem Internet herunterlädt. Virenscanner können den Bereich der Firmware eines USB-Geräts nicht überprüfen, sodass diese Angriffe bisher erfolgreich durchgeführt werden können.

2.4 Rechtliche Grundlagen

Die rechtlichen Grundlagen für den Einsatz technischer Hilfsmittel zum unbefugten Aufzeichnen oder Manipulieren von Daten sind differenziert zu betrachten, denn meist ist die Rechtmäßigkeit einer Verwendung fallbezogen. Der Paragraph §202a Strafgesetzbuch [stg] regelt den unbefugten Zugriff auf Daten folgender-

maßen:

> (1) Wer unbefugt sich oder einem anderen Zugang zu Daten, die nicht für ihn bestimmt und die gegen unberechtigten Zugang besonders gesichert sind, unter Überwindung der Zugangssicherung verschafft, wird mit Freiheitsstrafe bis zu drei Jahren oder mit Geldstrafe bestraft.
> (2) Daten im Sinne des Absatzes 1 sind nur solche, die elektronisch, magnetisch oder sonst nicht unmittelbar wahrnehmbar gespeichert sind oder übermittelt werden.

Dies bedeutet zum Bespiel, dass es verboten ist mithilfe eines Hardware-Keyloggers die Tastatureingaben einer anderen Person unbefugt aufzuzeichnen.

Auch einem Arbeitgeber ist es im Allgemeinen nicht gestattet, Daten der Arbeitnehmer ohne deren Wissen festzuhalten [bil]. Darüber hinaus regelt das Betriebsverfassungsgesetz, dass der Betriebsrat bei der Einführung technischer Hilfsmittel zur Aufzeichnung von Verhalten und Leistung der Arbeitnehmer Mitbestimmungsrechte besitzt [bet].

Anders verhält es sich beim Einsatz technischer Hilfsmittel zum Aufzeichnen von Daten bzgl. der Strafverfolgung. Zwar ist §100h Abs. 1 Nr. 2 Strafprozessordnung [stp] laut einer internen Einschätzung der Generalstaatsanwaltschaft München [Mü11] nicht ausreichend, jedoch wurde mit §20k BKA-Gesetz [bka] entsprechende Grundlagen für den Einsatz solcher Hilfsmittel geschaffen. Somit ist z.B. der Einsatz von “Remote Forensic Software” (ugs. “Bundestrojaner”) unter bestimmten Umständen möglich, welcher eine Funktion zur Aufzeichnung von Tastatureingaben besitzt [dI07].

Kapitel 3

Implementierung

Die Umsetzung der zu Beginn genannten Funktionalitäten wird innerhalb der folgenden Teilabschnitte beschrieben. Der erste Teilabschnitt dieses Kapitels dokumentiert die entwickelte Software [Fri14], welche den Mikrocontroller steuert und den Programmablauf der einzelnen Funktionalitäten zeigt. Der zweite Teil dieses Kapitels erläutert den Aufbau der Elektronik und zeigt wie die verwendete Hardware mit dem Mikrocontroller zusammengebaut wurde.

3.1 Softwaredokumentation

Die implementierten Softwarekomponenten [Fri14] gliedern sich in die drei Abschnitte für die jeweiligen Funktionalitäten, die Aufnahme von Tastatureingaben, die Wiedergabe von Tastatureingaben mittels SD-Karte und die Wiedergabe von Tastatureingaben über Ethernet. Diese beinhalten Hilfsfunktionen für die Tastatur, für die SD-Karte und für die Webseite, welche vorab in den nächsten Abschnitten beschrieben werden. Abschließend wird der gesamte Programmablauf beschrieben, der aus den Standardfunktionen des Arduino Mikrocontroller besteht und den Wechsel zwischen den Funktionalitäten ermöglicht.

3.1.1 Hilfsfunktionen für die Tastatur

3.1.1.1 void initKeys(int dataPin, int clockPin)

3.1.1.2 void setHigh(int pin)

3.1.1.3 void setLow(int pin)

3.1.1.4 unsigned char readKeys(int dataPin, int clockPin)

3.1.1.5 void sendKeys(int dataPin, int clockPin, unsigned char data)

3.1.1.6 void writeKeys(unsigned char data)

3.1.2 Hilfsfunktionen für die SD-Karte

Die implementierten Hilfsfunktionen für die SD-Karte werden benötigt, um sowohl die mitgelesenen Tastatureingaben abzuspeichern, als auch abgespeicherte Tastatureingaben wiederzugeben. Hierfür werden bestehende Funktionen aus der SD-Bibliothek von Arduino verwendet [ardb]. Zu beachten ist, dass Dateinamen nicht als String, sondern nur als Char-Array anzugeben sind.

3.1.2.1 void initCard(int sdPin)

Mit dieser Methode wird eine Verwendung einer SD-Karte ermöglicht, weshalb sie aufgerufen werden muss bevor eine der folgenden Hilfsfunktionen für die SD-Karte verwendet wird. Zuerst Dann wird gerprüft, ob die SD-Karte überhaupt vorhanden ist.

3.1.2.2 String readFile(char* filename)

Die Methode nimmt den Dateinamen einer Datei als Parameter entgegen und gibt einen String zurück, welcher den Textinhalt der Datei darstellt. Zunächst prüft die Methode, ob die Datei mit dem Namen existiert. Im Fall dass diese existiert, wird die Datei mit Leserechten geöffnet und solange sie verfügbar ist wird der Inhalt Zeichen für Zeichen ausgelesen und zu einem String zusammengefügt. Anschließend wird die Datei geschlossen und dieser String zurückgegeben. Falls die Datei nicht verfügbar ist, wird eine entsprechende Fehlermeldung auf der Konsole zurückgegeben.

Abbildung 3.1: Aktivitätsdiagramm für die Methode reader()

Abbildung 3.2: Aktivitätsdiagramm für die Methode writer()

3.1.2.3 void writeFile(char* filename, String content)

Mithilfe dieser Methode ist es möglich Daten in Form eines Strings in eine bestimmte Datei zu schreiben. Hierfür werden sowohl der Dateiname als auch der String dieser Funktion als Parameter übergeben. Die Datei mit dem übergebenen Namen wird mit Schreibrechten geöffnet und falls die Datei noch nicht existiert, wird sie automatisch erstellt. Anschließend wird überprüft, ob die Datei erfolgreich geöffnet werden konnte. Im Erfolgsfall wird der String an den bisherigen Inhalt der Datei angehangen und die Datei danach geschlossen. Andernfalls wird eine entsprechende Fehlermeldung auf der Konsole zurückgegeben.

3.1.2.4 void deleteFile(char* filename)

Diese Methode nimmt einen Dateinamen als Parameter entgegen und führt einen Löschungsbefehl für eine Datei mit diesem Dateinamen auf der SD-Karte aus. Anschließend wird geprüft, ob eine Datei mit dem Dateinamen noch existiert. Falls dies nicht der Fall ist, wird eine Nachricht ausgegeben, dass die Datei erfolgreich gelöscht wurde. Andernfalls wird eine Fehlermeldung ausgegeben.

Abbildung 3.3: Aktivitätsdiagramm für die Methode sender()

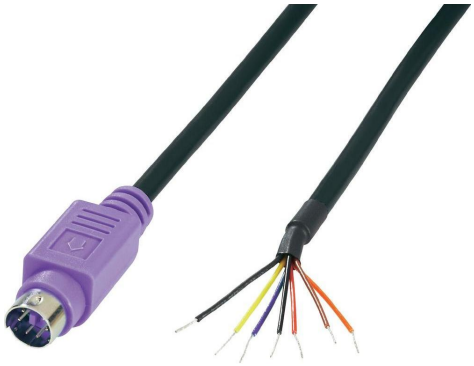

Abbildung 3.4: PS/2 Male

Abbildung 3.5: PS/2 Female

Abbildung 3.6: Schema des Aufbaus fritzing

3.1.3 Hilfsfunktionen für die Webseite

3.1.3.1 void sendWebsite(EthernetClient client, String serverState)

3.1.4 Aufnahme von Tastatureingaben

3.1.5 Wiedergabe von Tastatureingaben mittels SD-Karte

3.1.6 Wiedergabe von Tastatureingaben über Ethernet

3.1.7 Gesamter Programmablauf

3.2 Aufbau der Elektronik

Verwendete Kabel [ps2b] [ps2a], Arduino Mega und Ethernet [arda] und PS/2-Tastatur.

Abbildung 3.7: Foto des Arduino Ethernet Shields und des Mega 2560 Boards

Kapitel 4

Evaluation

Test der 3 Funktionalitäten Für Wiedergabe Test mit Windows und Linux zum Aufruf der Konsole (deutsches Tastaturlayout)

4.1 Abwehrmechanismen

Mit Host Befehlen testen, wie das Gerät reagiert [Mih10]

CTRL ALT T (wait) xdg-open http://www.google.de (wait) ENTER
14 f0 14 11 f0 11 2c f0 2c 00 00 00 22 f0 22 23 f0 23 34 f0 34 12 f0 12 4a f0 4a 44 f0 44 4d f0 4d 24 f0 24 31 f0 31 29 f0 29 33 f0 33 2c f0 2c 2c f0 2c 12 f0 12 49 f0 49 12 f0 12 3d f0 3d 12 f0 12 3d f0 3d 1d f0 1d 1d f0 1d 1d f0 1d 49 f0 49 34 f0 34 44 f0 44 44 f0 44 34 f0 34 4b f0 4b 24 f0 24 49 f0 49 23 f0 23 24 f0 24 29 f0 29 00 00 00 5a f0 5a

Tabelle 4.1: Linux Test

WINDOWS R (wait) CMD (wait) ENTER (wait) start http://www.google.de (wait) ENTER
e0 1f e0 f0 1f 2d f0 2d 00 00 00 21 f0 21 3a f0 3a 23 f0 23 00 00 00 5a f0 5a 00 00 00 1b f0 1b 2c f0 2c 1c f0 1c 2d f0 2d 2c f0 2c 29 f0 29 33 f0 33 2c f0 2c 2c f0 2c 12 f0 12 49 f0 49 12 f0 12 3d f0 3d 12 f0 12 3d f0 3d 1d f0 1d 1d f0 1d 1d f0 1d 49 f0 49 34 f0 34 44 f0 44 44 f0 44 34 f0 34 4b f0 4b 24 f0 24 49 f0 49 23 f0 23 24 f0 24 29 f0 29 00 00 00 5a f0 5a

Tabelle 4.2: Windows Test

Kapitel 5

Zusammenfassung

5.1 Ausblick

Literaturverzeichnis

[arda] Arduino Produkte. http://store.arduino.cc/index.php?main_page=index&cPath=11. Aufrufdatum: 23.09.2014.

[ardb] Arduino SD Library. http://arduino.cc/en/pmwiki.php?n=Reference/SD. Aufrufdatum: 23.09.2014.

[bet] Betriebsverfassungsgesetz §87 Mitbestimmungsrechte. http://www.gesetze-im-internet.de/betrvg/__87.html. Aufrufdatum: 23.09.2014.

[bil] Verordnung über Sicherheit und Gesundheitsschutz bei der Arbeit an Bildschirmgeräten (Anhang). http://www.gesetze-im-internet.de/bildscharbv/anhang_8.html. Aufrufdatum: 23.09.2014.

[bka] BKA Gesetz §20k Verdeckter Eingriff in informationstechnische Systeme. http://www.gesetze-im-internet.de/bkag_1997/__20k.html. Aufrufdatum: 23.09.2014.

[Bun12] Bundeskriminalamt. Cybercrime Bundeslagebild 2012. http://www.bka.de/nn_224082/SharedDocs/Downloads/DE/Publikationen/JahresberichteUndLagebilder/Cybercrime/cybercrimeBundeslagebild2012,templateId=raw,property=publicationFile.pdf/cybercrimeBundeslagebild2012.pdf, 2012. Aufrufdatum: 23.09.2014.

[Cha03] Adam Chapweske. The PS/2 Mouse/Keyboard Protocol. http://www.computer-engineering.org/ps2protocol/, 2003. Aufrufdatum: 23.09.2014.

[Che09] K. Chen. Reversing and exploiting an Apple firmware update. http://www.blackhat.com/presentations/bh-usa-09/

CHEN/BHUSA09-Chen-RevAppleFirm-PAPER.pdf, 2009. Aufrufdatum: 23.09.2014.

[dI07] Bundesministerium des Innern. Fragenkatalog der SPD-Bundestagsfraktion. http://netzpolitik.org/wp-upload/fragen-onlinedurchsuchung-SPD.pdf, 2007. Aufrufdatum: 23.09.2014.

[duc] USB Rubber Ducky. http://hakshop.myshopify.com/collections/usb-rubber-ducky/products/usb-rubber-ducky-deluxe. Aufrufdatum: 23.09.2014.

[Eng87] Stephen Engelberg. Embassy security: Story of failure. http://www.nytimes.com/1987/04/19/world/embassy-security-story-of-failure.html?pagewanted=all&src=pm, 1987. Aufrufdatum: 23.09.2014.

[fem] MiniDIN-6 Connector Pinout. http://commons.wikimedia.org/wiki/File:MiniDIN-6_Connector_Pinout.svg. Aufrufdatum: 23.09.2014.

[Fri14] Andreas Fritz. BadPS2: Implementierung der Bachelorarbeit. https://github.com/s6anfrit/badps2, 2014. Aufrufdatum: 23.09.2014.

[kee] Hardware Keylogger Vergleich. http://www.keelog.com/de/keylogger_comparison.html. Aufrufdatum: 23.09.2014.

[Kei05] Gregg Keizer. Keyloggers Foiled In Attempted $423 Million Bank Heist. http://www.informationweek.com/keyloggers-foiled-in-attempted-%24423-million-bank-heist/d/d-id/1031143?, 2005. Aufrufdatum: 23.09.2014.

[keya] Keylogger-hardware-PS2. http://commons.wikimedia.org/wiki/File:Keylogger-hardware-PS2.jpg#mediaviewer/File:Keylogger-hardware-PS2.jpg. Aufrufdatum: 23.09.2014.

[keyb] Keylogger-hardware-USB. http://commons.wikimedia.org/wiki/File:Keylogger-hardware-USB.jpg#mediaviewer/Datei:Keylogger-hardware-USB.jpg. Aufrufdatum: 23.09.2014.

[Kir08] Jeremy Kirk. Swedish Police Warn of Tampered Credit Card Terminals. http://www.pcworld.com/article/155525/article.html, 2008. Aufrufdatum: 23.09.2014.

[KN14] Jakob Lell Karsten Nohl, Sascha Krißler. BadUSB. https://srlabs.de/blog/wp-content/uploads/2014/07/SRLabs-BadUSB-BlackHat-v1.pdf, 2014. Aufrufdatum: 23.09.2014.

[Mih10] Fabian Mihailowitsch. Detecting Hardware Keyloggers. http://conference.hackinthebox.org/hitbsecconf2010kul/materials/D1T1%20-%20Fabian%20Mihailowitsch%20-%20Detecting%20Hardware%20Keyloggers.pdf, 2010. Aufrufdatum: 23.09.2014.

[Moc05] Bernward Mock. Die PS/2 Tastaturschnittstelle (Übersetzung). http://www.marjorie.de/ps2/ps2.pdf, 2005. Aufrufdatum: 23.09.2014.

[Mü11] Generalstaatsanwaltschaft München. Leitfaden zum Datenzugriff insbesondere für den Bereich Telekommunikation. http://cryptome.org/isp-spy/munich-spy-all.pdf, 2011. Aufrufdatum: 23.09.2014.

[pre] PrettyOS Kernel Keyboard. http://sourceforge.net/p/prettyos/code/HEAD/tree/trunk/Source/kernel/keyboard.c. Aufrufdatum: 23.09.2014.

[ps2a] Kabel PS/2 Female. http://www.exp-tech.de/Zubehoer/Steckverbinder/PS-2-Wired-Connector-Panel-Mount-MiniDIN-6.html. Aufrufdatum: 23.09.2014.

[ps2b] Kabel PS/2 Male. http://www.conrad.de/ce/de/product/601847/. Aufrufdatum: 23.09.2014.

[ps2c] PS2Keyboard Library. http://www.pjrc.com/teensy/td_libs_PS2Keyboard.html. Aufrufdatum: 23.09.2014.

[sca] Scancode-Set 2 Ausschnitt. http://retired.beyondlogic.org/keyboard/scancode.gif. Aufrufdatum: 23.09.2014.

[stg] Strafgesetzbuch §202a Ausspähen von Daten. http://www.gesetze-im-internet.de/stgb/__202a.html. Aufrufdatum: 23.09.2014.

[stp] Strafprozessordnung §100h. http://www.gesetze-im-internet.de/stpo/__100h.html. Aufrufdatum: 23.09.2014.

[TH08] Felix Freiling Thorsten Holz, Markus Engelberth. Learning More About the Underground Economy: A Case-Study of Keyloggers and Dropzones. https://ub-madoc.bib.uni-mannheim.de/2160/1/impersonation_attacks_TR.pdf, 2008. Aufrufdatum: 23.09.2014.

Anhang A

Anhang

A.1 PS/2-Tastatur Scancode-Set 2

Die folgenden Angaben sind Hexadezimalwerte für Tastaturen mit 101-, 102- oder 104-Tasten:

KEY	MAKE	BREAK	JavaScript
A	1c	f0 1c	65
B	32	f0 32	66
C	21	f0 21	67
D	23	f0 23	68
E	24	f0 24	69
F	2b	f0 2b	70
G	34	f0 34	71
H	33	f0 33	72
I	43	f0 43	73
J	3b	f0 3b	74
K	42	f0 42	75
L	4b	f0 4b	76
M	3a	f0 3a	77
N	31	f0 31	78
O	44	f0 44	79
P	4d	f0 4d	80
Q	15	f0 15	81
R	2d	f0 2d	82

S	1b	f0 1b	83
T	2c	f0 2c	84
U	3c	f0 3c	85
V	2a	f0 2a	86
W	1d	f0 1d	87
X	22	f0 22	88
Y	35	f0 35	89
Z	1a	f0 1a	90
0	45	f0 45	48
1	16	f0 16	49
2	1e	f0 1e	50
3	26	f0 26	51
4	25	f0 25	52
5	2e	f0 2e	53
6	36	f0 36	54
7	3d	f0 3d	55
8	3e	f0 3e	56
9	46	f0 46	57
‘	0e	f0 0e	192
-	4e	f0 4e	189
=	55	f0 55	187
\	5d	f0 5d	220
BKSP	66	f0 66	8
SPACE	29	f0 29	32
TAB	0d	f0 0d	9
CAPS	58	f0 58	20
L SHFT	12	f0 12	16
L CTRL	14	f0 14	17
L GUI	e0 1f	e0 f0 1f	91
L ALT	11	f0 11	18
R SHFT	59	f0 59	
R CTRL	e0 14	e0 f0 14	
R GUI	e0 27	e0 f0 27	92
R ALT	e0 11	e0 f0 11	
APPS	e0 2f	e0 f0 2f	93
ENTER	5a	f0 5a	13

ESC	76	f0 76	27
F1	05	f0 05	112
F2	06	f0 06	113
F3	04	f0 04	114
F4	0c	f0 0c	115
F5	03	f0 03	116
F6	0b	f0 0b	117
F7	83	f0 83	118
F8	0a	f0 0a	119
F9	01	f0 01	120
F10	09	f0 09	121
F11	78	f0 78	122
F12	07	f0 07	123
PRNT SCRN	e0 12 e0 7c	e0 f0 7c e0 f0 12	
SCROLL	7e	f0 7e	145
PAUSE	e1 14 77 e1 f0 14 f0 77	-none-	19
[	54	f0 54	219
INSERT	e0 70	e0 f0 70	45
HOME	e0 6c	e0 f0 6c	36
PG UP	e0 7d	e0 f0 7d	33
DELETE	e0 71	e0 f0 71	46
END	e0 69	e0 f0 69	35
PG DN	e0 7a	e0 f0 7a	34
U ARROW	e0 75	e0 f0 75	38
L ARROW	e0 6b	e0 f0 6b	37
D ARROW	e0 72	e0 f0 72	40
R ARROW	e0 74	e0 f0 74	39
NUM	77	f0 77	144
KP /	e0 4a	e0 f0 4a	111
KP *	7c	f0 7c	106
KP -	7b	f0 7b	109
KP +	79	f0 79	107
KP EN	e0 5a	e0 f0 5a	
KP .	71	f0 71	110
KP 0	70	f0 70	96

KP 1	69	f0 69	97
KP 2	72	f0 72	98
KP 3	7a	f0 7a	99
KP 4	6b	f0 6b	100
KP 5	73	f0 73	101
KP 6	74	f0 74	102
KP 7	6c	f0 6c	103
KP 8	75	f0 75	104
KP 9	7d	f0 7d	105
]	5b	f0 5b	221
;	4c	f0 4c	186
’	52	f0 52	222
,	41	f0 41	188
.	49	f0 49	190
/	4a	f0 4a	191

A.2 Befehlssatz

A.3 Quellcode

```
void setup() {

}

void loop() {

}
```

Listing A.1: Arduino