

Ein digitales Hilfssystem zur interaktiven Unterrichtspartizipation

Florian Grunert

Zabel-Gymnasium Gera

20. März 2019

Inhaltsverzeichnis

Kurzfassung	2
Einleitung	3
1 Geräte und Materialien	3
1.1 Verarbeitungseinheit	3
1.2 Eingebende Peripherie	4
1.3 Ausgebende Peripherie	4
2 Softwareeinrichtung	4
2.1 Grundkonfiguration	4
2.2 Server	5
2.3 Stream	6
2.4 Webinterface	8
2.5 Server-Interface	11
2.6 Touchscreen	12
3 Gehäuse	13
3.1 Material	13
3.2 Aufbau	13
Ausblick	14
Abbildungs-, Unterstützer- und Quellenverzeichnis	15

Kurzfassung

Ziel dieses Projektes ist es, Schülern, welche temporär oder permanent eine Bildungseinrichtung nicht besuchen können, eine Möglichkeit zu bieten, dennoch am Unterricht zu partizipieren.

Dies wird mit einen Livestream des Unterrichtsraumes, welcher von Kamera und Mikrofon vor Ort aufgezeichnet, verarbeitet und als Stream über das Webinterface bereitgestellt wird, realisiert. Jener ist auf jedem gängigen Webbrowser und Endgerät abrufbar. Mit diesem Webinterface ist es dem Schüler möglich, sich an Diskussionen oder Unterrichtsgesprächen aktiv zu beteiligen sowie Fragen zu aktuellen Unterrichtseinheiten stellen, sei es in Textform oder unter Zusendung von Bild- oder Dokumentdateien, beispielsweise seine schriftlichen Lösungsansätze beinhaltend. Andersherum kann der Lehrer dem Schüler häufig ohnehin bereits digital vorliegendes Unterrichtsmaterial zur Verfügung stellen, damit letztgenannter Arbeitsblätter oder audiovisuelle Inhalte gleichermaßen bearbeiten kann.

Die Verarbeitungseinheit stellt ein *Raspberry Pi* dar, ein Einplatinencomputer, welcher auf diese Art und Weise als ein Server fungiert. Auch bietet er selbst über den angeschlossenen Touchscreen eine Oberfläche, die zulässt, den Stream und die Verarbeitungseinheit zu steuern, welche in *Python* geschrieben wurde. Des Weiteren gibt jene Oberfläche Informationen zum Betriebszustand aus und zeigt aktuelle Nachrichten der Fernteilnehmer. Additiv zum Touchscreen werden wichtige Informationen auch auditiv wahrnehmbar durch Lautsprecher und Sprachsynthese ausgegeben.

Um aber eine Verknüpfung zwischen Web- und Serverinterface herzustellen, um Informationen auszutauschen, wird die serverseitige Skriptsprache PHP verwendet.

Geräte ähnlicher Funktion mögen bereits existieren, doch haben sie Preise um 2.500,00 € [2], was besonders dafür, die Bildung einem jeden zu ermöglichen, ein Preis ist, für den nicht jede Bildungseinrichtung oder gar Privatperson monetäre Mittel aufwenden kann. Das hier vorgestellte System liegt in einem Kostenbereich von circa 130,00 €.

Außerdem sind diese Produkte kommerziell; dieses ist quelloffen und für jeden nach seinen individuellen Bedürfnissen modifizierbar.

Einführung

Einigen Schülern ist es temporär oder permanent nicht möglich, eine Schule physisch zu besuchen, beispielsweise aufgrund von Erkrankungen. Daher ist Ziel der Arbeit, ein digitales, quelloffenes und kostengünstiges System zu entwickeln, durch welches ein interaktiver Austausch von Inhalten in beide Richtungen ermöglicht wird.

Das System übernimmt die physische Präsenz, da ein erheblicher Teil der Schüler über die aktive Verfolgung des Unterrichts effektiver lernt, als allein durch das spätere Lesen schriftlicher Aufzeichnungen: Sollen die Schüler zum Erlernen von Inhalten so vorgehen, behalten sie nach [1] nur etwa 30 % der Informationen, während sich Schüler, die dafür hören, sehen und darüber diskutieren, etwa 70 % der Inhalte merken. Zu 90 % Behaltensquote kommt es nur, wenn Schüler additiv zu den zu erlernenden Bereichen noch motorisch agieren, was allerdings ebenjenen Schülern mit körperlichen Einschränkungen ohnehin nur schwer möglich ist.

Der Unterricht wird den Schülern daher durch eine Ton- und Bildübertragung in Form von Tafelbildern, Erläuterungen, Arbeitsblättern und weiteren Inhalten online in Echtzeit zur Verfügung gestellt, damit diese in der Lage sind, jenen wie bei physischer Anwesenheit nachzuvollziehen und selbst mit dem Unterrichtsraum zu interagieren, beispielsweise um Lösungen von Aufgabenstellungen einzureichen sowie Probleme und Fragen anschaulich zu beantworten oder selbst zu stellen.

Besondere Anforderung an die technische Realisierung ist die Universalität: Auf den Ziel-systemen der Schüler wäre zum Beispiel eine Installation von Zusatzprogrammen eine Operation, für die nicht alle genügend technische Kenntnisse besitzen und möglicherweise über nicht ausreichend Speicherplatz verfügen. Dazu muss gewährleistet werden, dass die interaktive Teilnahme plattformunabhängig ist. Des Weiteren muss auch bezüglich des Einsatzorts des Geräts ausreichende Unabhängigkeit gewährt sein. Größere Veränderungen in der Netzwerkinfrastruktur der Lehranstalt wären hinderlich. So erscheint am einfachsten, einen Server zu realisieren, welcher entsprechende Eingabegeräte besitzt und deren Datenstrom live über ein Webinterface zur Verfügung stellt.

Die geringen Kosten können durch die Verwendung entsprechend günstiger Komponenten, wie z.B. des Kleincomputers *Raspberry Pi* sowie durch die Nutzung kostenfreier Software, welche im besten Fall quelloffen ist, um jene an das verwendete System zu adaptieren.

1 Geräte und Materialien

1.1 Verarbeitungseinheit

Wie bereits einleitend erwähnt, ist es notwendig, das System kostengünstig zu gestalten, um jedermann Bildungsmöglichkeiten zugänglich zu machen.

Dieses soll audiovisuelle Daten verarbeiten und zur Verfügung stellen sowie eine Weboberfläche zur Ansicht der Daten und zur Interaktion bieten. Weiterhin soll eine Portabilität gewährleistet sein. Insgesamt bedeutet das, dass eine kompakte, kleine Verarbeitungseinheit benötigt wird, deren Rechenleistung in keinem außergewöhnlichem Maß liegen muss, die aber ausreichende Konnektivität für die Eingabe- und Ausgabegeräte sowie für die Internetverbindung besitzen sollte.

Diese Vorgaben eignen besonders den Einplatinencomputer *Raspberry Pi* für das Projekt: Das Modell 3 B+ besitzt vier USB-Anschlüsse, 2.4 und 5 GHz WLAN, einen LAN-Anschluss, einen HDMI-Ausgang und einen Audio-Klinkenausgang, arbeitet mit 1 GB

RAM und vier Prozessorkernen auf 1.4 GHz. Zusätzlich ermöglichen sogenannte GPIO-Pins, programmierbare Ein- und Ausgänge, die Möglichkeit, über den Raspberry Pi mit anderen Systemen oder elektrischen Schaltungen zu kommunizieren oder diese zu steuern. Betrieben wird er mit 5 V Gleichspannung, welche jeder USB-Anschluss liefern kann [3]. Der Neupreis dieses Produkts, welches in Großbritannien produziert wird, liegt im deutschen Handel bei etwa 30,00 €.

Zum Betrieb wird eine mikro-SD-Karte benötigt, ausreichend sind 8 GB Speicherkapazität; eine solche Karte ist bereits ab 4,00 € im Handel erwerbbar.

1.2 Eingebende Peripherie

Notwendig für eine audiovisuelle Datenübertragung in Richtung der Schüler sind einerseits ein Mikrofon und andererseits eine Kamera.

Optimale Kompatibilität mit dem Raspberry Pi weist natürlich das dazugehörige Kameramodul auf, welches über einen eigenen Kameraanschluss auf dem Raspberry Pi mit selbigem verbunden wird. Es unterstützt Auflösungen bis 1080p bei 30 Bildern je Sekunde [4]. Der Neupreis im deutschen Handel beträgt ungefähr 11,00 €.

Ein Mikrofon kann per USB-Anschluss verbunden werden und ist ab 5,00 € in Deutschland kaufbar.

1.3 Ausgebende Peripherie

Auch am Standort des Verarbeitungsgeräts sollten Ausgabegeräte, zum Beispiel für Rückmeldung zum Betriebsstatus und zur Ausgabe der Schülerinteraktionen vorhanden sein. Ideal dafür ist ein Touchscreen, der per HDMI angeschlossen wird und Energie aus einem der USB-Ports erhält. Ein solcher mit einer Bildschirmdiagonale von 7" ($\approx 17,78$ cm) wird zwar von der Raspberry Pi Foundation produziert, allerdings besitzt dieser nur eine Auflösung 800 x 480 Pixeln bei einem Preis von circa 70,00 € bei Kauf in Deutschland [5]. In online-Kaufportalen kann man einen solchen kapazitiven Touchscreen auch ab 40,00 € erstehen, sogar mit einer Auflösung von 1024 x 600 Pixeln; dieser wird im Folgenden verwendet.

Ein Lautsprecher nahezu beliebiger Qualität und Größe gepaart mit einer zusammengelöteten Verstärkerschaltung ermöglichen auch eine Audioausgabe, deren Realisierungskosten marginal sind.

2 Softwareeinrichtung

Ziel des Gerätes ist es, ein kostengünstiges System bereitzustellen, dessen Kern gemeinfrei ist. Daher werden unter github.com/f-grunert/grit sämtliche relevanten Programmdateien sowie Konstruktionsschemata zur Verfügung gestellt, genau so wie eine Anleitung auf Deutsch und Englisch. Dadurch ist es auch international möglich, dieses Projekt in Kollaboration mit anderen Entwicklern über ein erstelltes Kommentarsystem weiter zu perfektionieren.

2.1 Grundkonfiguration

Der Raspberry Pi läuft optimal mit Raspbian, einem Derivat der Linux-Distribution Debian, welches auf die Hardware des Raspberry Pis angepasst wurde. Nach dem Schreiben des

Abbilds auf eine Mikro-SD-Karte und Anschluss aller Peripheriegeräte kann der Raspberry Pi gestartet werden. Möchte man jenen per SSH fernbedienen, so ist zuvor eine leere Datei mit dem Namen `ssh` auf der boot-Partition der Mikro-SD-Karte zu erstellen.

Grundsätzlich sollten vor der Nutzung eines Raspberry Pis einige Konfigurationen vorgenommen werden: Über das *Software Configuration Tool* (*raspi-config*) muss aus Sicherheitsgründen das ursprüngliche Kennwort des Standardbenutzers *pi*, welches zuvor *raspberry* lautete, geändert werden.

Außerdem ist es dort möglich, optional die WiFi-Konnektivität einzurichten. Es ist zu empfehlen, dort einzustellen, der Raspberry Pi möge mit dem Starten warten, bis eine Netzwerkverbindung aufgebaut ist.

Ferner muss das Interface, welches beim Start geladen werden soll, das graphische und die Anmeldung die automatische sein, was ebenso in dem Tool aktivierbar ist.

Elementar ist dort auch die Aktivierung des Kamera-Interfaces genau so wie die Forcierung des Audio-Ausgangs durch die Klinkenbuchse, weil der Ton über die dort angeschlossenen Lautsprecher wiedergegeben werden soll.

Eine Ausweitung des Dateisystems, damit die gesamte Speicherkapazität der Speicherkarte genutzt werden kann, wird empfohlen.

Vor jeglicher Installation von Paketen sind grundsätzlich die Paketquellen zu aktualisieren. Vor der Benutzung des Geräts ist abzuwägen, welche Pakete der mitgelieferten Software tatsächlich für eine interaktive Unterrichtsverfolgung benötigt werden und welche nicht. Im Grunde sind die Java-IDEs *BlueJ* und *Greenfoot*, die IDE *Geany*, das Mathematik-Softwarepaket *Mathematica*, das Entwicklungswerkzeug *Node-RED*, die Programmiersprache *Scratch* und ihre IDE, den Hardware-Emulator *Sense HAT*, die Python-Entwicklungsumgebung *IDLE*, das Musikprogramm *Sonic Pi*, die Python-IDE *Thonny*, das Logiksimulationsprogramm *SmartSim*, die Bürosoftware *Libreoffice*, der E-Mail Client *Claws Mail*, der Remotesoftware *VNC*, das Spiel *Minecraft*, die minimalistischen Webbrowser *dillo* und *Epiphany* sodass diese Pakete einfach entfernt werden können. Um die vorhandenen Pakete aktuell zu halten, sollten diese anschließend aktualisiert werden. Ein darauf folgender Neustart ist nicht zwingend notwendig, aber angeraten:

```
1  sudo apt --purge remove bluej geany greenfoot wolfram-engine nodered
   scratch scratch2 sense-hat sense-emu-tools idle idle3 sonic-pi
   python3-thonny smartsim libreoffice* claws-mail realvnc-vnc-server
   realvnc-vnc-viewer minecraft-pi dillo epiphany-browser
2  sudo rm -r /etc/vnc
3  sudo apt --purge autoremove
4  sudo apt dist-upgrade
5  sudo reboot now
```

Damit direkt beim Systemstart erkennbar ist, ob alle Gerätekomponenten ordnungsgemäß funktionieren, ist zudem ein Selbsttest auf python-Basis realisiert worden, der partiell die Funktionen der GPIO-Eingänge nutzt, welche die digitalen Werte 0 und 1 einerseits senden, andererseits aber auch empfangen können.

2.2 Server

Das Interaktionsinterface sollte möglichst auf jedem System laufen und keine Installation auf selbigem benötigen. Ideal ist daher eine Realisierung über einen Server, der von den

Schülern aufgerufen werden kann.

Geeignet erscheint dafür die Webserver-Software *nginx*, die im Gegensatz zur Standardsoftware für Server *Apache* auch mit geringerer Hardwareleistung eine gute Performance erreicht.

nginx selbst kann unter Raspbian einfach mit installiert werden. Weil dazu auch PHP benötigt wird, um serverseitige Austausch zu ermöglichen, wird dieses gleich mit installiert:

```
sudo apt install nginx php-fpm
```

Das Arbeitsverzeichnis des Webserver findet sich nun unter `/var/www/html`. Dessen Bestandteile sind nun auf dem Raspberry Pi unter `http://localhost` und im Netzwerk des Einplatinencomputers unter dessen IP sichtbar.

Um das `index.php`-Skript automatisch zu laden, wenn keine Datei in der URL spezifiziert ist, ist in der Konfigurationsdatei von *nginx* unter `/etc/nginx/sites-enabled/default` mit root-Rechten wie folgt zu bearbeiten:

```
43 # Add index.php to the list if you are using PHP
44 index index.html index.htm index.nginx-debian.html index.php;

54 # pass PHP scripts to FastCGI server
55 #
56 location ~ /\.php$ {
57     include snippets/fastcgi-php.conf;
58     #
59     #    # With php-fpm (or other unix sockets):
60     fastcgi_pass unix:/var/run/php/php7.0-fpm.sock;
61     #    # With php-cgi (or other tcp sockets):
62     #    fastcgi_pass 127.0.0.1:9000;
63 }
```

Danach sollte das Verzeichnis `/var/www/html` so bearbeitet werden, dass dortige Änderungen auch vom Standardbenutzer vorgenommen werden können. Final ist der Server neu zu laden.

```
1 sudo chmod -R 777 /var/www/html
2 sudo /etc/init.d/nginx reload
```

Damit Zugriffe nur von autorisierten Personen stattfinden können, wurde eine Abfrage von Benutzername und Passwort implementiert, die entweder auf die schulinterne Nutzerdatenbank oder auf eine dynamisch veränderbare, bereits erstellte Datenbank zugreift. Außerdem ist eine SSL-Verschlüsselung vom Ausgangsserver über das Schulnetzwerk bis zum heimischen Endgerät gewährleistet.

2.3 Stream

Integral für eine digitale Teilnahme am Unterricht ist natürlich ein Stream, der die Bild- und Tonsignale überträgt. Dieser Bereich muss in zwei Teile zerlegt werden: Einerseits in die Aufzeichnung und Dekodierung der Inhalte und andererseits in die Bereitstellung dieser auf dem Server.

Online steht für beide Bereiche ein Grundgerüst bereit, dass allerdings selbst kompiliert werden muss.

Zunächst sind dafür jedoch einige Voraussetzungen zu erfüllen: Erst muss der Treiber für die Raspberry-Kamera installiert werden. Lässt man sich die vorhandenen Gerätedateien mittels `ls /dev` anzeigen, so erscheint dort noch nicht die Videokamera unter `video0`. Dazu muss nach [6] der Name des Treibers `bcm2835-v4l2` an die Datei `/etc/modules` mit root-Rechten angehängt und das Gerät neu gestartet werden. Nun sollte die Kamera in `ls /dev` angezeigt werden.

Zum ersten Teil ist `picam` zu installieren. `picam` enthält ein Programmsammlung, die es ermöglicht, aus Audio- und Videoaufnahmegegeräten einen H.264/AAC encodierten MPEG-TS-Stream zu erstellen, einerseits im RTMP-Format, andererseits im HLS-Format. Da für die Einbettung in den Server und Wiedergabe von RTMP-Streams wenig gute Software existiert, exklusive ein Stream an YouTube, welcher aber Latenzen von bis zu 45 s hervorruft, ist der HLS-Stream zu präferieren; hierzu gibt es ein zu `picam` gehöriges Programm, welches den Stream mit einer Latenz von drei bis 4 Sekunden bereitstellen kann.

Um `Picam` zu installieren, sind nach [7] als erstes einige Ordner zu erstellen. Des Weiteren ist empfehlenswert, symbolische Verknüpfungen zum Ordner `/run/shm` zu erstellen, weil dann die entstehenden temporären Daten nicht auf die Speicherkarte, sondern in den Arbeitsspeicher geschrieben werden, was kürzere Zugriffszeiten und weniger Zugriffe auf die Mikro-SD-Karte bedeutet:

```
1 mkdir -p /run/shm/rec
2 mkdir -p /run/shm/hooks
3 mkdir -p /run/shm/state
4 mkdir -p ~/picam/archive
5
6 ln -sfn ~/picam/archive /run/shm/rec/archive
7 ln -sfn /run/shm/rec ~/picam/rec
8 ln -sfn /run/shm/hooks ~/picam/hooks
9 ln -sfn /run/shm/state ~/picam/state
```

Um nun die entsprechende Binärdatei, das eigentliche `picam`-Programm auszuführen, muss dessen Archiv zunächst heruntergeladen und entpackt werden:

```
1 wget https://github.com/iizukanao/picam/releases/download/v1.4.7/
   picam-1.4.7-binary.tar.xz
2 cp picam-1.4.7-binary/picam ~/picam/
3 tar xvf picam-1.4.7-binary.tar.xz
4 rm -r picam-1.4.7-binary.tar.xz picam-1.4.7-binary
```

`picam` könnte nun gestartet werden, doch benötigt das Programm den Namen des Audioaufnahmegegeräts. Dieser lautet bei einem USB-Mikrofon am Raspberry Pi in der Regel `hw:1,0`. Überprüft werden kann dies mit dem folgenden Befehl:

```
1 arecord -l
2 > **** Liste der Hardware-Geräte (CAPTURE) ****
3 > Karte 1: Device [USB PnP Sound Device], Gerät 0: USB Audio [USB
   Audio]
4 > Sub-Geräte: 1/1
5 > Sub-Gerät #0: subdevice #0
```

Die Bezeichnung des Mikrofons setzt sich im Fall eines USB-Geräts wie folgt zusammen:
hw:<Kartenummer>,<Gerätenummer>.

Ziel ist aber nicht, Aufnahme zu tätigen, sondern den Kamerastream auf dem Server wiederzugeben. Dazu muss zunächst der Serversoftware der Ort des Streams bekannt gemacht werden; dies geschieht durch die Änderung der Datei `/etc/nginx/sites-available/default` mit root-Rechten:

```
48 location /hls/ {
49     root /run/shm;
50 }
```

Anschließend ist der Server über

```
sudo service nginx restart
```

neu zu starten, damit die Änderungen übernommen werden.

In das Serververzeichnis `/var/www/html` kann nun der `picam-viewer` aus [8] kopiert werden, nachdem zuvor die Standardbegrüßungsseite von `nginx` gelöscht wurde:

```
1 rm index.nginx-debian.html
2 sudo git clone https://github.com/kamranayub/picam-viewer.git
   /var/www/html
```

Ruft man nun auf dem Raspberry Pi `http://localhost` oder im lokalen Netzwerk die IP-Adresse des selbigen auf, so ist zunächst sichtbar, dass kein Stream läuft. Er muss erst gestartet werden, nämlich über:

```
/home/pi/picam/picam -o /run/shm/hls - --channels 1 --audiobitrate 96000
  --videobitrate 4000000 --vfr --avclevel 3.1 --autoex --alsadev hw:1,0
```

Die Bedeutung dieser Parameter ist grundsätzlich selbsterklärend; jene Werte wurden durch eine Testphase gefunden. Weitere Erläuterungen dazu finden sich in [7].

Ist der Mikrofonpegel zu hoch oder zu niedrig, kann jener entweder über das graphische Einstellungsmenü oder über das dazugehörige Kommandozeilenprogramm `alsamixer` verändert werden.

2.4 Webinterface

Weil allerdings diese Ausgabe hinsichtlich Größe und Gestaltung etwas suboptimal ist und auch PHP-Inhalte eingebunden können werden sollen, wird eine PHP-Datei mit dem Namen `index.php` im HTML-Ordner erstellt.

In diese wird per `<iframe>` die `index.html`-Datei eingebunden. Des Weiteren wird darin eine Javascript-Funktion erstellt, die im Abstand von zwei Sekunden über das Senden eines `XMLHttpRequest()` überprüft, ob eine bestimmte temporäre Datei `stream.tmp` im zu erstellenden Unterordner `interaction` vorhanden ist, die vom Interface auf dem Übertragungsgerät erstellt wurde. Ist dies der Fall, so bedeutet dies, dass der Stream aktiv ist und er über JavaScript eingeblendet werden kann.

Weiterhin sind auf der `index.php`-Seite formularähnliche Elemente, wie Bereiche zum

Stellen von Fragen, Serviceanfragen, optional mit Upload einer Bilddatei oder einem Dokument eingebaut, die einen `XMLHttpRequest()` an eine Verarbeitungs-PHP-Datei beim Absenden auslösen, welche die übertragenen Inhalte an eine LOG-Datei weitergibt. Damit zwischenzeitlich keine anderen Schreibzugriffe auf diese Datei stattfinden können, beispielsweise durch das Benutzerprogramm auf dem Server, wird vor jedem Schreibzugriff eine temporäre Datei erstellt, deren Existenz geprüft wird und die so den Zugriff sperrt. Letztere wird regelmäßig vom Interface-Programm auf dem Server abgefragt und dann bei neuem Inhalt entsprechende Aktionen ausgeführt.

Da es aber sehr unüblich ist, bei einem vom Schüler gesendetes Bild zum Abruf in eine LOG-Datei zu schreiben, wird in der Datei nur der Speicherort des hochgeladenen Bildes mitgeteilt. Zu beachten ist hierbei noch, dass python zur Einbindung in das GUI nur GIF-Dateien zulässt, weshalb z.B. Dokumente vorher konvertiert werden müssen, was durch eine Implementation realisiert wurde. Ist ein USB-Stick am Raspberry angeschlossen, geschieht nach dem Upload durch den Schüler automatisch eine Kopie auf diesen Stick, damit der Lehrer auch im Nachhinein damit arbeiten und z.B. des Schülers Ergebnisse verfasster schriftlicher Ausarbeitungen evaluieren kann.

Beim Aufruf der Seite wird additiv geprüft, ob im Serververzeichnis ein bestimmter symbolischer Link vorhanden ist. Sollte dem so sein, werden die Dateien dieses Linkziels in die Website eingebunden und sind für den Schüler herunterladbar.

Ist dem nicht so, überprüft das PHP-Skript, ob ein externes Speichermedium angeschlossen ist, was einbindbar wäre und erstellt in diesem Falle den symbolischen Link dazu.

Bezüglich des Aussehens der Oberfläche wird über die Einbindung einer globalen CSS-Datei ein einheitliches Design gewährleistet. Da der Fokus bis dato auf Funktionalität lag, ist deren Reichhaltigkeit allerdings derzeit noch eher mäßig. Dennoch ist aktuell schon eine Responsivität gegeben: Der Inhalt der Website passt sich durch entsprechend gestalteten HTML-Code der Bildschirmgröße der Endgeräte an, zumindest bei den gängigsten Mobiltelefonen.

System zur interaktiven Unterrichtspartizipation

Livestream:



Interaktionen:

Frage stellen:

Name:

Frage:

Datei (optional):

Keine Datei ausgewählt.

Lehrerdateien:

- [entwicklung-der-quantenphysik.pdf](#)
- [unterricht.pptx](#)
- [AB.docx](#)

Serviceanfrage:

Name:

Anfrage:

▾

Abbildung 1: online-Benutzeroberfläche

2.5 Server-Interface

Besagtes Interface ist ein Programm, welches in *Python* mit *Tkinter* für das GUI geschrieben wurde. Hierbei war wichtig, dass das Programm selbsterklärend und einfach bedienbar ist.

Daher startet dieses gleich nach dem Hochfahren des Raspberry Pis im Vollbildmodus, wo ebenso Buttons für das Starten und Beenden der live-Übertragung, für den Aufruf des Einstellungsmenüs, ebenso wie Ausgabebereiche für die Nachrichten der Webnutzer und für Systemstatusmeldungen vorhanden sind. Eine Ausrichtung und Kontrolle optimal eingestellter Kamerawerte wird durch einen entsprechenden Bereich im Fenster vorgenommen, der ein sich im einsekündigen Abstand aktualisierendes Bild der Kamera über eine Endlosschleife wiedergibt. Dies wird über einen Unterprozess mit dem Modul `subprocess` über `subprocess.Popen()` realisiert, indem damit eine Schleife aufgerufen wird, die regelmäßig ein neues Bild aufnimmt und in der Anwendung aktualisiert.

Die Kamerawerte werden im System durch die Modifikation vorhandener Konfigurationsdateien von `picam` eingestellt. Mit dem Befehl

```
os.mknod('~/.picam/hooks/wb_off')
```

des `os`-Moduls kann zum Beispiel eine entsprechende Konfigurationsdatei zur Regelung des Weißabgleichs im `picam`-Ordner erstellt werden, die den Abgleich deaktiviert. Der Dateiname nach dem Unterstrich fungiert hierbei in der Regel als Parameter.

Sendet ein Schüler über das Netzwerk eine Anfrage, sei es ein Unterrichtsbeitrag ggf. mit Bild/Dokument oder einen Servicewunsch, so wird zunächst eine visuelle und eine kurze auditive Ausgabe ausgelöst. Werden dann zu gegebener Zeit von der Lehrkraft die entsprechenden Knöpfe angewählt, wird die Anfrage eingeblendet und ein evtl. vorhandenes Bild/Dokument vergrößert. Da ein ständiges Beobachten des Bildschirms von allen Schülern während des Unterrichts hinderlich wäre, nutzt das Interface auch die angeschlossene Audioausgabe. Selbige besteht aus einem simplen Lautsprecher, welcher über eine verlötete, eigens entworfene Verstärkerschaltung, die mit dem Klinkeausgang des Raspberry Pis verbunden ist, diesen über `subprocess.call()` über den entsprechenden Kommandozeilenbefehl ansteuert. `call()` wird hier verwendet, weil nicht wie bei `Popen()` die entsprechende Funktion im Hintergrund aufgerufen wird, sondern der weitere Programmverlauf angehalten wird. Grund dafür ist, dass die Ansage zu Ende gesprochen werden soll, bevor ggf. eine nächste verarbeitet werden kann. Mit dem synthetischen Sprachausgabeprogramm für die Konsole mit dem Namen *pico2wave* werden wichtige Informationen so auch auditiv wahrnehmbar bei entsprechendem Tastendruck ausgegeben. Eine Wiedergabe erfolgt nicht direkt nach Erhalt der Frage, weil diese Partizipationsweise den Unterricht beeinträchtigen könnte; beispielsweise ist es von der Lehrkraft nicht in allen Momenten des Unterrichts gewollt, dass Schüler Fragen stellen. Außerdem wäre dann eine Bevorzugung des „digitalen Schülers“ gegeben, was so verhindert werden kann.



System läuft fehlerfrei

Interaktionen:

Karlheinz > Welche weiteren Möglichkeiten der Lösung eines Gleichungssystems existieren?

Stream starten

System herunterfahren

Beiträge (1)

Serviceanfragen (3)

Einstellungen

Stream angehalten

Abbildung 2: Benutzeroberfläche des Servers

Um beispielsweise den Systemstatus darzustellen werden auch die GPIO-Pins, die jeweils die Stati `high` und `low` kennen, wobei ersterer einer Spannung von ca. 3,3 V und letzterer einer Spannung von 0 V entspricht. Damit ist es möglich, eine farbige RGB-LED anzu- steuern, unter Python mit dem `RPi.GPIO`-Modul oder direkt in der Raspbian-Shell mit dem Programm `pigpio`.

Damit grundlegende Aktionen, wie z.B. das temporäre Deaktivieren, das Ausschalten oder das Neustarten ohne größere Menüführung ausgeführt werden können, wurde ein entsprechender Taster eingebaut, der dies haptisch erleichternd ermöglicht.

2.6 Touchscreen

Der Touchscreen ist nicht von Beginn an mit dem Raspberry Pi nutzbar; zunächst müssen entsprechende Treiber installiert werden. Dazu wird erneut in den Ordner `/home/pi/src` gewechselt und dort des Herstellers Treiber-GitHub-Verzeichnis geklont und in jenes ge- wechselt. Darauf ist der Treiber über die entsprechende Datei zu installieren; das System wird darauf neu gestartet. [9]

Bei Bedarf kann danach noch das Kalibrierungsprogramm `xinput-calibrator` und eine Bild- schirmtastatur wie `onboard` installiert werden:

- 1 `git clone https://github.com/goodtft/LCD-show.git`
- 2 `cd LCD-show`
- 3 `sudo bash LCD7C-show`
- 4 `sudo apt install xinput-calibrator onboard`

Jene besitzt auch eine einstellbare `auto-hide`-Funktion, nachdem die Tastatur bei Ein- gabefeldern automatisch erscheinen soll, was in der Praxis aber nicht immer reibungslos funktioniert.

3 Gehäuse

Da das Projekt für jeden zugänglich sein wird, ist es jedem möglich, innerhalb seiner handwerklichen Mittel und Fähigkeiten ein Gehäuse für Peripherie sowie Verarbeitungseinheit zu erstellen. Zur Erstellung des Prototyps durch den Verfasser dieser Arbeit wird im Folgenden jedoch zumindest eine Empfehlung erörtert oder die eigene Vorgehensweise dargestellt.

3.1 Material

Im Allgemeinen kann als Material ein beliebiges verwendet werden. Trotzdem ist auf elektrisch leitende zu verzichten, denn im Falle eines technischen Defekts, beispielsweise beim Lösen eines Kabels, kann daraus ein Kurzschluss mit anderen Geräteteilen entstehen, die dadurch ggf. zerstört werden könnten.

Möchte man einen guten Kompromiss zwischen Stabilität und Einfachheit der Verarbeitung eingehen, so stellt Holz eine gute Alternative dar. Hier besteht die Möglichkeit, sogar auf einfachste Art und Weise eine Gestaltung durch Bemalung vorzunehmen. Dabei können sich die Schüler auf spielerische Art und Weise mit Aussehen und Bedienung des Geräts auseinandersetzen.

Ebenso Kunststoff, der exempli causa als Acrylglas einen Einblick in das Innenleben des Geräts ermöglichen kann oder mit einem 3D-Drucker ohne größere Arbeit angefertigt werden kann, eignet sich als Gehäusematerial.

3.2 Aufbau

Das System muss so aufgebaut sein, dass die Kamera eine Position einnimmt, die sich ungefähr auf Höhe der Tafel und der unterrichtenden Lehrkraft befindet. Ein durch Schrauben verstellbarer Dreibein-Ständer sowie eine Holzplatte mit -umrandung darauf, auf bzw. in der der Kasten mit Verarbeitungseinheit und Peripherie platziert wird, stellen eine Konstruktion dar, die eine Zweckmäßigkeit darstellt. Die exakten Modelldaten dazu stehen gemeinfrei auf GitHub zur Verfügung.

Ausblick

Mit meinem Gerät zur interaktiven Unterrichtspartizipation ist es möglich geworden, Schülern, die den Unterricht nicht besuchen können, zu ermöglichen, dennoch an selbigen zumindest digital von Zuhause aus teilzunehmen: Sie können Fragen stellen, in Text- und Bildform, Antworten erhalten und den Unterricht live mit Ton und Bild verfolgen.

Der Server stellt die dazugehörige Oberfläche den Abwesenden, welche diese über einen handelsüblichen Browser aufrufen kann, zur Verfügung. Das geschieht jetzt noch in einer optisch eher weniger ansprechenden Art und Weise. Ziel in der Zeit bis zum Wettbewerb ist es, dies für den Nutzer etwas ansprechender zu gestalten und die Responsivität zu verbessern, die bis dato nicht bei allen Endgeräten gegeben ist.

Weiterhin besteht auf der Seite der sich im Unterricht befindlichen Personen das Potential, das Gerät auch zu Recherchezwecken geringen Umfangs oder zur Wiedergabe audiovisueller Beiträge zu nutzen. Dazu kann die Anwendung weiter optimiert werden, ebenso die Systemoberfläche mit dem Webbrowser. Die Verwendung der Bildschirmtastatur hierbei gestaltet sich bisher auch eher umständlich, sodass angedacht ist, diese nicht über die bereits vorhandene auto-hide-Funktion zu verwenden, sondern über einen Button, der jene ein- und ausblendet.

Bereits angesprochen wurde die Möglichkeit einer Dokumentenkamera. Inwieweit dies tatsächlich nötig ist und inwiefern realisierbar, weil der Raspberry Pi selbst nur einen Kameraanschluss besitzt, ist zu untersuchen. Unter anderem findet dazu eine Erprobungsphase am Zabel-Gymnasium Gera und ggf. an der Dualen Hochschule Gera-Eisenach stattfinden und entsprechende Verbesserungsvorschläge der Nutzer überprüft und gegebenenfalls angewendet.

Zudem könnte in der Theorie das gesamte System auch auf einem USB-Stick mit installiertem Debian eingebettet werden, sodass dazu nicht zwangsläufig das physikalische Gerät nachgebaut, sondern ein mit Standardkomponenten ausgestatteter Laptop verwendet werden kann.

Abbildungs-, Unterstützer- und Quellenverzeichnis

- [1] KILP, ELÓIDE: Spiele für den Fremdsprachenunterricht: Aspekte einer Spielandragogik, Tübingen 2003, S. 27
- [2] MDR THÜRINGEN: Roboter hilft krebskrankem Schüler. <https://www.mdr.de/thueringen/mitte-west-thueringen/erfurt/roboter-unterricht-krebskranker-schueler-100.html> (14.01.2018)
- [3] RASPBERRY PI FOUNDATION: Raspberry Pi 3 Model B+. <https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/#buy-now-modal> (08.12.2018)
- [4] RASPBERRY PI FOUNDATION: Camera Module V2. <https://www.raspberrypi.org/products/camera-module-v2/> (08.12.2018)
- [5] RASPBERRY PI FOUNDATION: Raspberry Pi Touch Display. <https://www.raspberrypi.org/products/raspberry-pi-touch-display/#buy-now-modal> (08.12.2018)
- [6] DONAUER, JÜRGEN: Kameramodul des Raspberry Pi, V4L-Treiber (Video4Linux) und motion. <https://www.bitblokes.de/kameramodul-des-raspberry-pi-v4l-treiber-video4linux-und-motion/> (10.12.2018)
- [7] IZUKA, NAO: iizukanao/picam. <https://github.com/iizukanao/picam> (16.12.2018)
- [8] IZUKA, NAO: iizukanao/picam. <https://github.com/iizukanao/picam> (04.01.2019)
- [9] o.A.: Driver installation instruction. In: o.A.: Raspberry Pi Touch Display, o.J.
- [abb1] GRUNERT, FLORIAN (2019): online-Benutzeroberfläche. Eigene Darstellung
- [abb2] GRUNERT, FLORIAN (2019): Benutzeroberfläche des Servers. Eigene Darstellung
- [Unterstützer] KAISER, THOMAS: Gabe von nützlichen Tipps bei der Erstellung der Arbeit