

Entwicklerdokumentation

Useless Box mit Emotionen

WS23/24, Smart Technology

Melvin Göttsche, Lukas Krämer, Mario Stötzel

Einführung

Als Erstsemesterprojekt haben wir uns dazu entschieden eine sogenannte Useless Box zu bauen. Diese besteht im Wesentlichen aus einem Schalter, der nach Umlegen durch den Nutzer mithilfe eines Servo Motors wieder in seine Ausgangsposition gebracht wird.

Die Useless Box ist zwar wenig sinnvoll, allerdings bietet sie sich insbesondere als erstes Projekt an, da sie sowohl elektronische als auch mechanische und softwareseitige Aspekte beinhaltet. Dieses Dokument soll einen guten Überblick über die Hardware sowie Software der Box vermitteln und dabei helfen, das Projekt gegebenenfalls nachzubauen, zu erweitern oder reparieren zu können.

Überblick Hardware

Die Hardware der Useless-Box wurde auf Grundlage der verschiedenen Eigenschaften, die zu Beginn des Projektes für die Box ausgewählt wurden, festgelegt. Bei den Materialien haben wir uns für einfache Leimholzplatten entschieden. Diese Leimholzplatten haben eine Stärke von 4mm, was trotz des geringen Gewichtes die Stabilität, aber auch die Mobilität und Einfachheit der Useless-Box aufrechterhält. Die Box hat die Außenmaße von 22 x 12 x 12 cm (L/H/B), ist einfach zusammengesetzt und besteht aus einer Kiste mit einem zweigeteilten Deckel.

Wir haben eine besondere Eigenschaft hinzugefügt, mit der die Box mit Hilfe eines Displays Emotionen in Form von digitalen Gesichtern anzeigen kann. Um dieses umzusetzen, besitzt die eine Hälfte des Deckels der Useless-Box eine Vertiefung, die mit einer durchsichtigen Plexiglasscheibe abgedeckt wird, um das darunter befindlichen Displays zu schützen. Außerdem wurde auf dieser Seite der Box ein zusätzliches Loch ausgeschnitten, um einen Kippschalter für die Hauptfunktion der Box unterzubringen. Die andere Hälfte des Deckels ist einfach gestaltet und hat nur die Funktion einer Klappe.

In der Box selbst wurden für den Hauptmechanismus eine Halterung für einen Servo-Motor sowie mit der Grundplatte verleimte Standfüße zur Befestigung der Platine montiert.

Wesentliche Teile des Gehäuses, welche die Wände, den Boden, die Standfüße der Platine und Teile der Halterung für den Servo-Motor beinhalten, wurden als dauerhafte Verbindungen mit Leim befestigt. Andere Teile jedoch wie der Motor und die Platine wurden verschraubt, um eine Wartung zu ermöglichen. Die Plexiglasabdeckung wurde mit Silikon befestigt, um die Anzeige des Displays nicht zu beeinträchtigen. Die Klappe im Deckel wurde mit Scharnieren montiert.

Der Arm des Servo-Motors der Useless-Box ist das einzige Teil, welches aus Plastik in einem 3D-Drucker hergestellt wurde. Die Holzelemente und Plexiglasэлеmente wurden gelasert und nur einzelne Teile zusätzlich gebohrt. Alle Holzkomponenten wurden außerdem grundiert, geschliffen und dann in schwarz lackiert.

Zur Vorbereitung der Projektumsetzung haben wir zudem mit Hilfe einer Software und in Pappe ein 3D Model des Grundprinzips der Box entworfen, an welchem später noch Optimierungen bis zum finalen Produkt gefunden werden konnten.

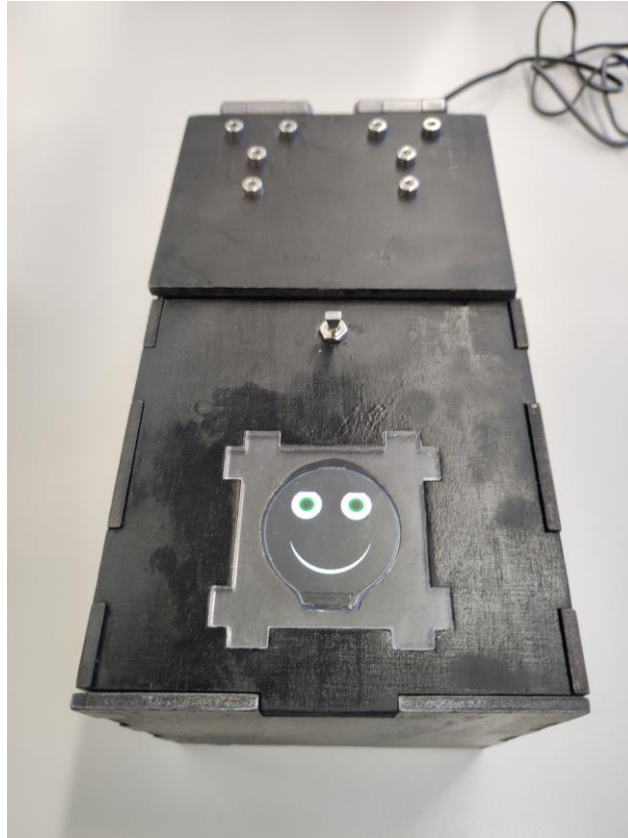


Bild 1: fertige Box

Überblick Elektronik

Die Elektronik unserer Useless Box sollte in der Lage sein, einen Servo anzusteuern (für das Umlegen des Arms), einen Schalter auszulesen und dem Benutzer in Form eines Displays Feedback geben zu können. Angebots hat sich daher die Nutzung eines Arduino Nano Mikrocontrollers, da er nicht nur unsere Anforderungen erfüllt, sondern wir durch private Projekte auch schon Erfahrung mit dem Board gesammelt haben und so die Entwicklung beschleunigt werden konnte.

Neben einem gewöhnlichen Schalter zum Umlegen, verwenden wir noch einen HiTec Servo, der den Arm bewegt, um später den Schalter umlegen zu können. Außerdem haben wir uns für ein rundes Display entschieden, was sich für das Darstellen von einem digitalen Gesicht als günstig erwies. Damit die Elektronik zuverlässig funktioniert und die Entwicklung nicht durch schwer reproduzierbare Wackelkontakte erschwert wird, haben wir uns von Anfang an dazu entschieden, für die gesamte Elektronik eine Leiterplatte zu erstellen.

Diese wurde zunächst in EasyEDA (Bild 2 und 3) designt und anschließend gefräst (Bild 4). Nach der Bestückung und kurzem Rauchtest haben wir nacheinander die Komponenten eingebaut und die Subsysteme der Reihe nach getestet.

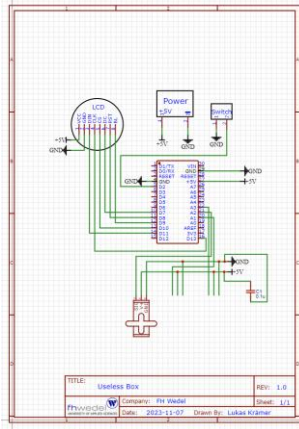


Bild 2: Schaltplan

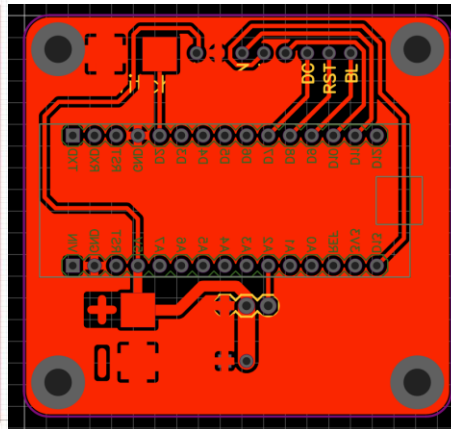


Bild 3: Platine

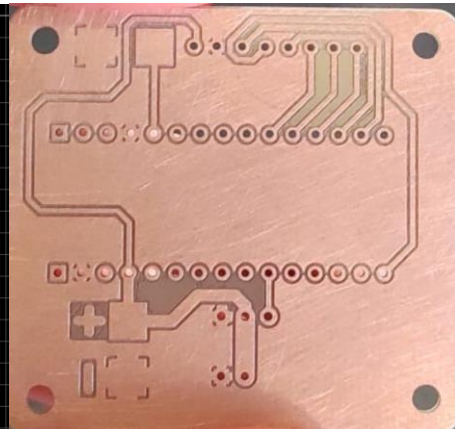


Bild 4: gefräste Platine

Überblick Software

Die Box fängt damit an den Zähler auf dem Display anzuzeigen, wie oft der Schalter bereits umgelegt worden ist. Sie zeigt diese Zahl so lange an, bis der Schalter umgelegt wird. Hiernach wird der Arm ausgefahren, bis der Schalter umgelegt wurde. Dies wird genutzt, um die Position des Schalters intern zu speichern. Damit sich die Box "merken" kann, wie oft der Schalter bereits umgelegt wurde, schreiben wir einen Zähler (unsigned, vier bytes groß) nach jedem Umlegen in die EEPROM des Arduinos. Da eine einzelne EEPROM Speicherzellen lediglich 100.000-mal beschrieben werden kann (100.000 stehen als "garantiert" im Datenblatt, unabhängige Tests ergaben deutlich höhere Zahlen) und wir dafür sorgen möchten, dass unsere Box auch so lange wie möglich korrekt funktioniert, haben wir uns dazu entschieden, ein rollierendes System zu verwenden, was im Folgenden näher erläutert wird. Das Abspeichern in die EEPROM erfolgt byteweise, also könnte man in Adresse 0 das LSB schreiben und die anderen drei Bytes in Adressen 1, 2 und 3. Da das aber nach 100.000 Schreibzyklen nicht mehr garantiert funktionieren würde, werden Byte 1, 2 und 3 in die Adressen 16, 17 und 18 geschrieben. Das LSB, was sich am häufigsten ändert, wird hingegen in einer Adresse zwischen 0 und 15 abgespeichert. Wird der Code initial auf dem Arduino ausgeführt, befindet sich in den Adressen 0 bis 15 überall eine 0, der Arduino wählt nun per "Zufallsgenerator" ($\text{millis} \bmod 16$) eine der Adressen aus und schreibt dort eine 1. Beim nächsten Schalterumlegen sucht er jetzt in den Adressen zwischen 0 und 15 die Adresse, deren Wert größer als 0 ist, folglich wird wieder in dieselbe Adresse wie zuvor geschrieben, bis der Wert 256 bzw. 0 erreicht wird. Dort wird dann erneut eine zufällige Adresse ausgewählt, um das LSB des Zählers abzuspeichern, die sich sehr wahrscheinlich (93,75%) von der letzten Adresse unterscheidet. Dadurch verschleißt die EEPROM deutlich langsamer und kann zumindest Zähler bis 1,6 Millionen garantiert abspeichern. Natürlich kann dieses System beliebig erweitert werden, sodass beispielsweise die ersten 32 oder 64 Adressen fürs Rollieren genutzt werden, was die Langlebigkeit weiter steigern würde.

Die Hauptmethode der Useless Box ist der Loop des Arduinos. Dieser wurde mit einer while Schleife getaktet, um die Animationen besser vorherzusehen. Sollte der Schalter in der Off-Position stehen, wird eine Variable happiness bis zu einem bestimmten Wert erhöht. Sollte der Schalter umgelegt worden sein und es läuft noch keine Animation, wird eine Animation zufällig ausgewählt, der Zähler für die Gesamtzahl

der Schalterumlege erhöht und die happiness um einen Wert reduziert. Sollte der Schalter vom Nutzer wieder zurückgelegt worden sein, während eine Animation noch läuft, so wird die Animation abgebrochen und der Arm zurückgefahren. Sollte eine Animation noch laufen, so wird eine Funktion aufgerufen, die wiederum die Funktion für die Animation direkt aufruft. Die Animationen werden in Steps aufgeteilt, damit die Main "zeitgleich" nebenbei laufen kann. Der erste Step jeder Animation ist es, den Servo zu attachen. In jedem Step wird, nachdem der Step abgeschlossen ist ein Zähler erhöht, welcher repräsentiert, welcher Step gerade dran ist. Um den Arm zu bewegen kann eine Funktion verwendet werden, der man sagen kann, wo der Arm hinsoll und in wie vielen Schritten er dort ankommen soll. Hierfür bestimmt die Funktion beim ersten Aufruf wie weit sich der Arm bei jedem Aufruf bewegen soll und gibt true zurück, wenn die finale Position erreicht worden ist.

Sollte im Moment keine Animation aktiv sein, wird das Gesicht auf dem Display aktualisiert. Hierbei wird die Variable happiness betrachtet und das Gesicht wird dementsprechend glücklich oder unglücklich gezeichnet. Bei jedem x-ten Aufruf der Hauptfunktion wird außerdem eine Funktion aufgerufen, die die Box blinzeln lässt, um sie humaner rüberkommen zu lassen.

Was wir nächstes Mal besser machen würden

Für unser nächstes Projekt sollten wir von Anfang an den Aspekt der einfachen Wart- und Reparierbarkeit denken, da bei uns aktuell viele Teile dauerhaft fest gelehmt oder nur schwer erreichbar sind. Mit genug Zeit lässt sich zwar jedes Teil tauschen, allerdings ist dies teils mit hohem Aufwand verbunden.

Wartungstätigkeiten

Zwar ist es schwierig, nach lediglich so kurzer Nutzung potentielle "Point of Failures" aufzuführen, wahrscheinlich wird jedoch als Erstes unser einziges bewegliches Teil, der Servo, versagen. Dieser kann getauscht werden, indem man zunächst die "Serviceklappe" hochhebt, die Servohalterung abschraubt und anschließend den Servo selbst aus seiner Halterung entfernt. Nach Einsetzen des neuen Servos, sollte die finale Servoposition anhand der beiden großen Schrauben, welche die Servohalterung mit der Box verbinden adjustiert werden, damit die Hand den Schalter der Box korrekt umlegen kann.

Weiterentwicklung

Was die Weiterentwicklung des Projektes betrifft, haben wir über das Ergänzen von RGB Leds nachgedacht, die zusätzlich zum Display die Emotionen der Box darstellen können. Diese könnte entweder von außen an die Box angebracht werden oder aber in Form von RGB-Streifen von innen in die Box geklebt werden, sodass sie durch die Klappe der Box scheinen, sobald der Servo diese hochhebt. Außerdem könnte eine SD-Karte mit verschiedenen Soundeffekten samt kleinem Lautsprecher ergänzt werden, um die Nutzung der Box noch interessanter zu gestalten. Abgespielt werden können dabei zum Beispiel Soundeffekte beim Umlegen des Schalters (bspw.: stöhnen, wenn die Box genervt ist). Zu guter Letzt können die von uns erstellen Animationen softwareseitig natürlich beliebig erweitert oder

abgeändert werden, was die Nutzung der Box sicherlich interessanter machen würde. Die Weiterentwicklung des Codes darf natürlich gerne in Form einer Pull-Request in unser Repository ergänzt werden.

Einrichten der Entwicklungsumgebung

Um an der Box weiterarbeiten zu können, muss zunächst die [Arduino IDE](#) heruntergeladen und installiert werden. Außerdem werden noch die beiden Libraries "[Adafruit GFX](#)" und "[Adafruit GC9A01A](#)" benötigt. Damit das Abspeichern des Schalterzählers in die EEPROM wie erwartet funktioniert, sollten außerdem die ersten 19 Adressen (also 0 bis 18) mit einer 0 beschrieben werden, dies erfolgt über den Befehl [EEPROM.write\(Adresse, 0\)](#).

Der gesamte Sourcecode samt weiter Informationen ist in unserem [GitHub Repository](#) zu finden.

Teileliste

- [Arduino Nano](#)
- [Waveshare 1.28inch Round LCD Module Display](#)
- [Servo HS-645MG](#)
- [5V Netzteil \(2.5A\)](#)
- [Buchse zum Netzteil](#)
- [Schalter](#)
- 2x Scharnier 35mm*125mm*1,5mm mit M4 Löchern
- [100 µF Elektrolyt-Kondensator](#)
- [8x Senkkopfschraube M4 10mm](#)
- 8x Schraube M4 14mm
- 16x Mutter M4
- 8x Schraube M3 10mm
- 8x Mutter M3
- 2x Unterlegscheibe M5
- 2x Unterlegscheibe M6
- 2x Schraube M5 45mm
- 1420cm² Leimholzplatten, 4mm dick (ohne Bohrungen)
- Holzblock 72 cm²
- 1m 0.75mm² Kupferkabel
- Holzleim
- Spraydose Grundierung
- Spraydose Scharz (Matt)
- 28cm² Plexiglas
- Silikon
- 2x Schraubenniete M5

Werkzeug

- Mini USB Kabel, zum Hochladen von Code auf den Arduino
- Lötkolben
- Lötzinn
- Abisolierzange
- Entsprechende Schraubendreher
- Schraubzwingen, um gelemte Teile zu fixieren
- Lasercutter
- 3D Drucker
- Kartuschenpresse
- Schleifpapier
- Feilen
- Nietenzange