

Projekt-Dokumentation: IOT-Steckuhr

Victor André Hourie Costa, Alexander Balleisen, Marco Schichtel, Jan Nowara

Tuesday 8th August, 2017

Contents

1	Idee	3
2	Technische Details	3
2.1	RFID-Modul	3
2.2	NTP-Server	3
2.3	Google-Sheet	3
3	Aufgetretene Probleme	4
3.1	Einfügen von Einträgen in das Google-Sheet	4
3.2	Gleicher RESET-Pin bei zwei Modulen	4
4	Betriebsanleitung	4
4.1	Setup	4
4.2	Neuen Benutzer registrieren	4
4.3	Benutzer trägt seine Zeit ein	5

1 Idee

Die Idee hinter unserem Projekt ist es eine IoT-Stechuhr (**IoT** = **I**nternet **o**f **T**hings) zu bauen. Diese Stechuhr soll es Mitarbeitern eines Unternehmens erlauben bei Arbeitsbeginn und Ende mit Hilfe einer RFID-Karte den genauen Zeitpunkt einzutragen, welcher dann am Ende eines Arbeitstages in einer online verfügbaren Tabelle dokumentiert wird. Dafür wird außer der eigentlichen Stechuhr noch ein Server benötigt der die Daten annimmt sowie eine Möglichkeit der Stechuhr die momentane GMT-Zeit mitzuteilen. Als kleines Gimmick zeigt die Stechuhr auch konstant die aktuelle Zeit bis auf die Sekunde genau an und kann somit auch als reguläre Uhr verwendet werden.

2 Technische Details

Verwendete Komponenten:

- 1 Arduino Mega 2560
- 1 Breadboard
- 1 Ethernet Shield
- 1 RFID RC522 MODULE + RFID-Karten
- 1 16x2 LCD Bildschirm
- 3 Buttons + Widerstände

2.1 RFID-Modul

Das RFID-Modul wird dazu benötigt um eine Identifizierungsmöglichkeit der Benutzer zu bieten in Form von RFID-Chips die sich durch den am Arduino angeschlossenen RFID-Reader als eindeutige ID auslesen lassen.

Das Kommunikationsprotokoll zwischen Arduino und dem RFID-Modul ist SPI. Um die RFID-Chips auszulesen wird im Programm-Code die *RFID.h*-Library verwendet.

2.2 NTP-Server

Der NTP-Server (NTP = **N**etwork **T**ime **P**rotocol) bietet externes Zeitmanagement an. Dafür schickt der Server eine Unix-Timestamp an den Client und das Protokoll hilft dabei den Delay der Nachricht zu berechnen. Das erlaubt eine einheitliche Zeitmessung auf mehreren verteilten Systemen zu haben anstatt die Zeitwerte der einzelnen Systeme von der Prozessorzeit des Systems abhängig zu machen. In unserem Fall verwenden wir den öffentlichen NTP-Server pool.ntp.org.

Weitere Informationen zu NTP können unter https://en.wikipedia.org/wiki/Network_Time_Protocol gefunden werden.

2.3 Google-Sheet

Das Google-Sheet wird dazu verwendet um die Arbeitszeiten der einzelnen User zu speichern und auszulesen. Dazu enthält es die folgenden vier Spalten:

- *DATE*: Enthält das Datum an dem der Benutzer gearbeitet hat
- *NAME*: Enthält den Benutzernamen
- *TIME_IN*: Enthält die Zeit des Arbeitsbeginns
- *TIME_OUT*: Enthält die Zeit des Arbeitsendes

Ein Spreadsheet kann unter <https://docs.google.com/spreadsheets> angelegt werden.

Um neue Einträge im Google-Sheet anzulegen wird ein HTTP-Request vom TELNET-Server *server.py* an script.google.com geschickt. Dieser Request hat das Format [https://script.google.com/macros/s/\(Script-ID\)/exec?name=\(Username\)&in=\(Arbeitsbeginn\)&end=\(Arbeitsende\)](https://script.google.com/macros/s/(Script-ID)/exec?name=(Username)&in=(Arbeitsbeginn)&end=(Arbeitsende)) und führt das Script *script_google.js* aus.

3 Aufgetretene Probleme

3.1 Einfügen von Einträgen in das Google-Sheet

Um das notwendige Skript auszuführen wird HTTPS benötigt, der Arduino supportet allerdings nur HTTP und alle anderen kostenlosen API-Service waren dafür nicht leistungsfähig genug. Daher haben wir einen in Python geschriebenen TELNET-Server als Layer zwischen Stechuhr und dem Google-Interface geschoben der die vom Arduino geschickten Werte parsed und sie zu dem entsprechenden HTTPS-Request konvertiert und an script.google.com weiterleitet.

3.2 Gleicher RESET-Pin bei zwei Modulen

Das Ethernet-Shield und das RFID-Modul benötigen den gleichen RESET-Pin, daher mussten wir die *RFID.h*-Library verwenden, da es die einzige Library ist, die es erlaubt den RESET-Pin für das RFID-Modul zu konfigurieren.

4 Betriebsanleitung

4.1 Setup

Voraussetzungen:

- Python Version 2.7 installiert
- Stechuhr nach dem Schaltplan zusammengebaut

Szenario:

1. Anlegen eines Google-Spreadsheets mit den entsprechenden Spalten (**WICHTIG:** Das Sheet muss für alle editierbar sein damit das Skript daran Änderungen vornehmen kann)
2. Anpassen der Dokument-ID im *script_google.js*
3. Erstellen eines Skriptes unter script.google.com und dort *script_google.js* einfügen.
4. Anpassen der Skript-ID in *server.py*
5. Starten des TELNET-Servers *server.py*
6. Ändern der Server-IP in *clock_system.ino* auf die Server-IP des TELNET-Servers
7. Starten des Arduinos

4.2 Neuen Benutzer registrieren

Voraussetzung:

- Stechuhr befindet sich im Lese-Modus.
- Der Benutzer besitzt einen noch nicht registrierten RFID-Chip.

Szenario:

1. Benutzer drückt den *Change State*-Button.
2. Stechuhr fordert den Benutzer dazu auf seinen RFID-Chip an den Reader zu halten.
3. Benutzer hält den Chip an den Reader.
4. Stechuhr fordert den Benutzer dazu auf den RFID-Chip zu bestätigen durch erneutes Einscannen.
5. Benutzer hält erneut den Chip an den Reader.
6. Stechuhr bestätigt erfolgreiche Validierung und geht zur Namenseingabe über.
7. Benutzer verwendet den *Change Letter*-Button um einen Buchstaben aus dem Alphabet auszuwählen und den *Enter*-Button um die Buchstaben-Auswahl zu bestätigen und um zum nächsten Buchstaben überzugehen.

8. Wenn die Namenseingabe vollständig ist bestätigt der Benutzer durch doppeltes Drücken des *Enter*-Buttons.
9. Die Stechuhr zeigt eine Bestätigungsmeldung an, trägt den neuen Angestellten. ein und geht wieder in den Lese-Modus über.

4.3 Benutzer trägt seine Zeit ein

Voraussetzung:

- Stechuhr befindet sich im Lese-Modus.
- Der RFID-Chip des Benutzers ist im System registriert.

Szenario:

1. Der Benutzer hält seinen RFID-Chip an den Reader um seinen Arbeitsbeginn einzutragen.
2. Die Stechuhr zeigt den Benutzernamen und den Zeitpunkt des Arbeitsbeginnes an.
3. Nach Arbeitsende hält der Benutzer seinen Chip erneut an den Reader.
4. Die Stechuhr zeigt den Benutzernamen und den Zeitpunkt des Arbeitsendes an.
5. Am Ende des Tages schickt die Stechuhr die Arbeitsbeginn- und Endzeiten der eingetragenen User an den Server wo sie dann in das Google-Sheet eingetragen werden.