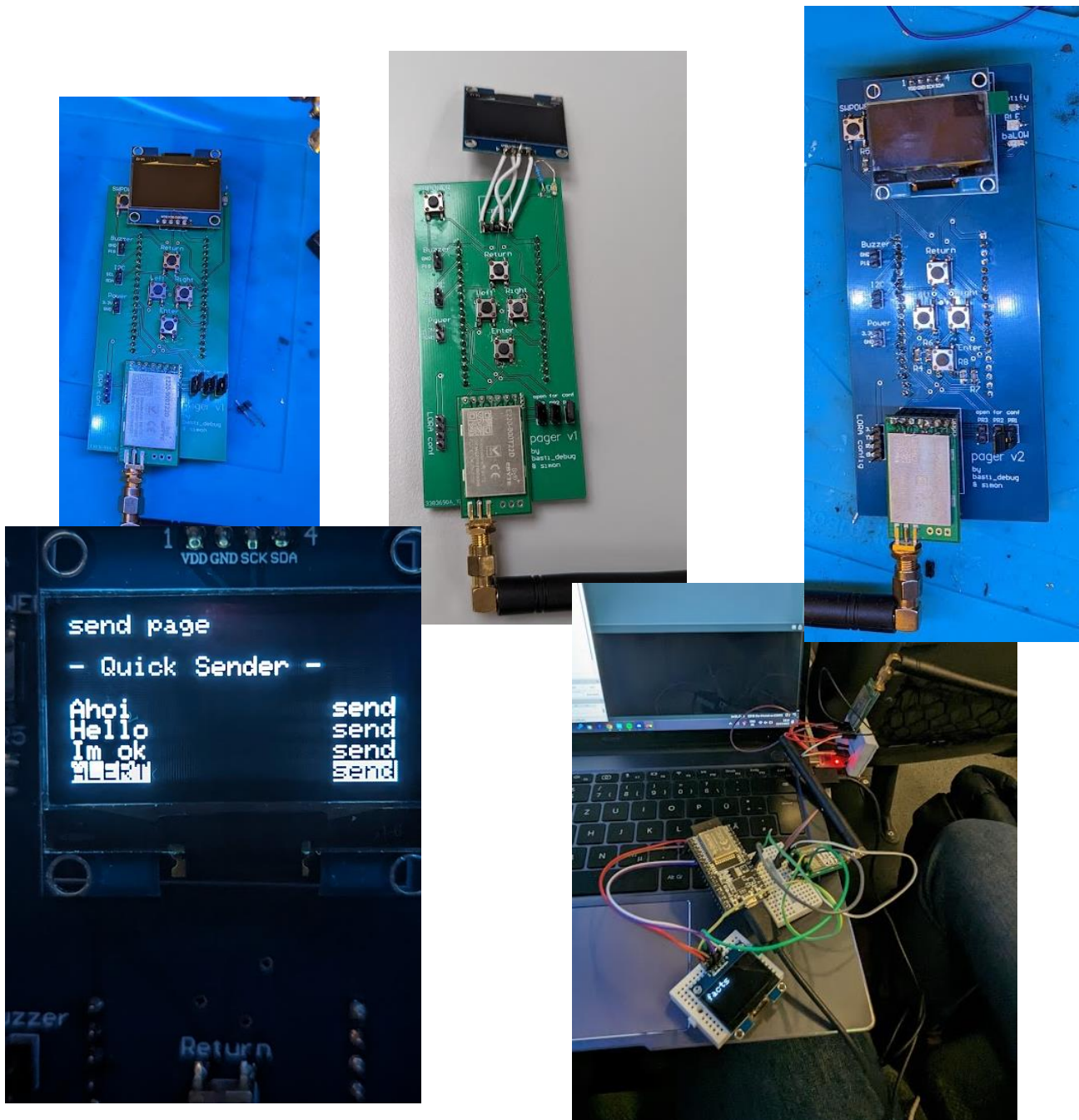


Projekt: LoRa - Handlicher Pager



Name Schüler: Koller Simon, Mayrhofer Sebastian

Betreuer: Prof. Jena

HTL Rankweil

4CHEL

Schuljahr 2022-23

Aufgabenbeschreibung:

Aus 2 LoRa Modulen soll ein Projekt unserer Wahl realisiert werden. Es handelt sich dabei um LoRa Module der e220-er Reihe von eByte. Die Kommunikation mit den Modulen verläuft seriell über UART. Die einzige Vorgabe, die wir einzuhalten haben, ist, dass eine Datenübertragung über die LoRa Module im Mittelpunkt des Projektes steht.

Zielsetzung:

Wir haben uns vorgenommen mithilfe der LoRa Module zwei handliche Pager zu realisieren. Dafür wollen wir ein ESP32 DevModule, Screens und Knöpfe verwenden. Als Ziel setzen wir uns mithilfe der Knöpfe zwischen Menüs hin und her zu wechseln und vorgefertigte Nachrichten zu schicken. Das Ganze soll kabellos sein, also Batterie-betrieben. Dennoch wollen wir auch individuelle Nachrichten mittels Laptop oder Handy schicken, damit die Möglichkeiten nicht begrenzt sind. Damit dies immer noch kabellos möglich ist, nehmen wir uns vor den Pager auch über Bluetooth steuerbar zu machen.

Inhaltsverzeichnis:

| | |
|--|---|
| Aufgabenbeschreibung: | 2 |
| Zielsetzung:..... | 2 |
| Inhaltsverzeichnis: | 2 |
| Liste Entwicklungstools: | 3 |
| Grundkonzept: | 3 |
| Funktionalität: | 3 |
| Blockdiagramme: | 3 |
| Technische Realisierung:..... | 4 |
| LoRa-Modul: | 4 |
| Einführung: | 4 |
| Einstellmöglichkeiten: | 5 |
| GUI: | 5 |
| Home-Page:..... | 5 |
| Send page: | 5 |
| Connection settings: | 6 |
| Erweiterte Anzeigeelemente: | 6 |
| Bedienelemente: | 6 |
| Akustische Empfangssignalisierung: | 6 |
| Piezo-Buzzer: | 6 |
| Desktop Applikation:..... | 7 |
| ESP32: | 7 |
| LoRa-Ansteuerung:..... | 7 |
| Screen-Ansteuerung: | 7 |
| Ablauf der Erstellung:..... | 8 |
| Probleme: | 8 |
| LORA Konfiguration | 8 |
| Screens..... | 8 |
| Bluetooth in WinUI3 | 8 |

| | |
|----------------------------|---|
| Bedienungsanleitung: | 8 |
| Literaturnachweise: | 9 |

Liste Entwicklungstools:

Microsoft Word: Version 2210

Greenshot: Version 1.2.10

Altium Designer: Version 23.3.1

Arduino IDE: Version 2.1.0

Git

eByte E220 LoRa Konfigurationstool

Grundkonzept:

Funktionalität:

Schnellnachrichten zwischen den Pagern über Menüs senden oder individuelle Nachrichten über angeschlossene Geräte (z.B. Laptop). Für kurze Strecken können Nachrichten über Bluetooth zwischen Laptop und Pager ausgetauscht werden. Bluetooth kann am Pager im Einstellungsmenü ein- bzw. ausgeschaltet werden und der Pager kann vor anderen Bluetooth-Geräten versteckt werden.

Blockdiagramme:



Kurzstreckenübertragung

per Bluetooth



Bluetooth-Gerät

Technische Realisierung:

LoRa-Modul:

Einführung:

LoRa ... Long Range

Als LoRa wird die von Semtech hergestellte Low-Power-Wide-Area-Network-Technology (LPWAN) bezeichnet.

Die großen Vorteile von LoRa sind bereits im Namen ersichtlich: große Reichweite und geringer Energieverbrauch.

Zur Signalübertragung verwendet LoRa die Chirp-Spread-Spectrum-Technologie (CSS). CSS kombiniert 2 Aspekte der Informationstechnik:

- Chirps:
- Spread-Spectrum:

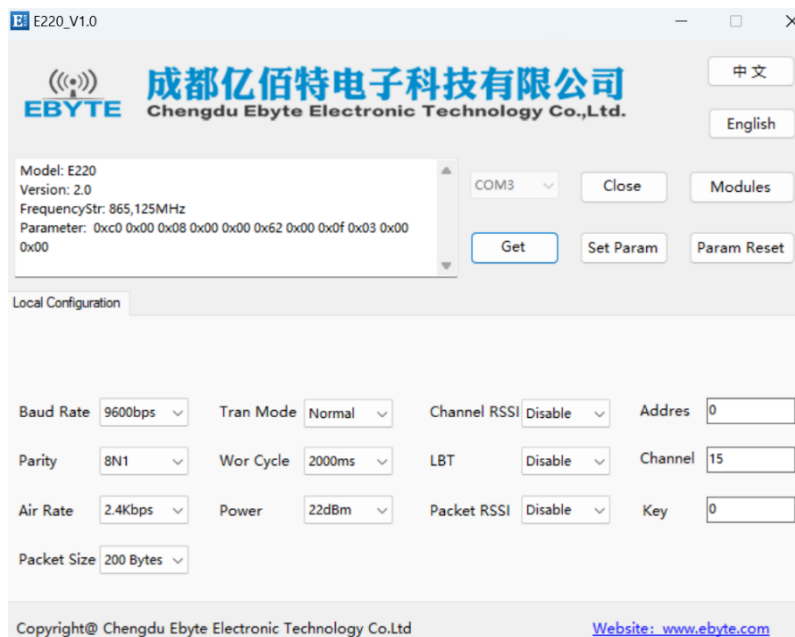
Chirps beschreiben Signale mit zeitlichen Frequenzänderungen, während Spread-Spectrum ein Verfahren ist, bei dem ein Signal gezielt umgewandelt wird, um eine größere Bandbreite und ein breiteres Frequenzspektrum zu nutzen.

Vorteile:

- Geringere Störanfälligkeit
- Hohe Datensicherheit
- Erschwert dritten, übertragene Daten auszulesen

Einstellmöglichkeiten:

Mit der Software von eByte (s.o.) kann das LoRa Modul konfiguriert werden:



Die wichtigsten Parameter dabei sind Baud Rate, „Adresse“ und Channel. Damit zwei oder mehrere LoRa-Module miteinander kommunizieren können, ist es wichtig, dass diese auf dieselbe Adresse und denselben Channel konfiguriert werden. Die Baud Rate von 9600bps haben wir gewählt damit wir auch einfach mit Terminal Programmen arbeiten können. Die restlichen Parameter können auf den Standardeinstellungen gelassen werden.

GUI:

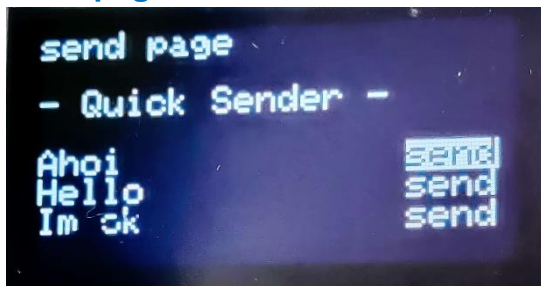
Wir verwenden das 1,3 Zoll I2C Display von AZ-Delivery mit einer Auflösung von 128 X 64 Pixel. Der Bildschirm dient als Benutzeroberfläche für den Pager. Hier kann der Benutzer die zuvor empfangenen Nachrichten oder die aktuellen Einstellungen ansehen. Außerdem kann er vorgefertigte Schnellnachrichten auswählen und verschicken. Das Ganze kann in 3 Schnellmenüs erreicht werden:

- Home-Page
- Send page
- Connection settings

Home-Page:

Bild und Beschreibung einfügen.

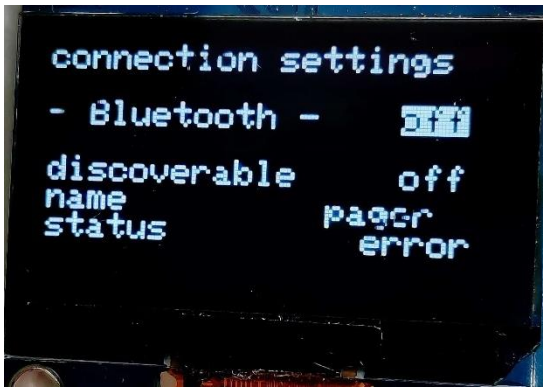
Send page:



Auf der send-Seite können die Schnellnachrichten ausgewählt und über LoRa versendet werden. Es besteht eine programmierbare Auswahl an Schnellnachrichten.

(Bild: 3 Schnellnachrichten „Ahoi“, „Hello“, „Im ok“. Ersteres ist ausgewählt)

Connection settings:



Hier können Bluetooth-Einstellungen vom Pager verändert werden. Zum einen kann Bluetooth ein- bzw. ausgeschaltet werden. Und es kann eingestellt werden, ob der Pager von anderen Bluetooth-Geräten erkannt werden kann.

Außerdem wird in der Info-Anzeige in der unteren Hälfte des Displays der Bluetooth-Name und der Status angezeigt.

(Bild: Status = error, weil noch nicht zu Gänze programmiert).

Erweiterte Anzeigeelemente:

Auf der rechten Seite des Displays gibt es 3 Leds:

- die erste, die Notification LED, war eigentlich vorgesehen, dass diese blinkt sobald eine neue Nachricht ankommt doch wir hatten leider Pins ohne Ausgangstreiber des ESPs geroutet und nun haben wir keinen Pin mehr übrig -> Funktion nicht implementiert
- die zweite, die BLE LED, leuchtet sobald Bluetooth aktiviert ist
- die dritte, die baLOW LED, war anfangs als Batterie Alarm vorgesehen, doch da wir die Batterie direkt in unseren Circuit einspeisen und kein BMS dazwischen ist, können wir nicht herausfinden ob die Batterie voll oder leer ist, wir wissen nur dass sie leer ist wenn der ESP abstürzt -> Daher haben wir diese als ALERT (Alarm) LED umfunktioniert, sie soll nun den Nutzer, zusammen mit dem Buzzer auf eine wichtige Nachricht hinweisen.

Bedienelemente:

Die Menüs werden über 5 Taster bedient. Mit den Taster „Left“ und „Right“ kann zwischen den Menüs gewechselt werden. In welche Richtung ist dabei selbsterklärend. Mit dem unteren Taster („Return“) kann gescrollt werden. Und mit „Enter“ wird die jeweilige Aktion ausgeführt. Z.B.: Bluetooth einschalten oder Schnellnachricht versenden.

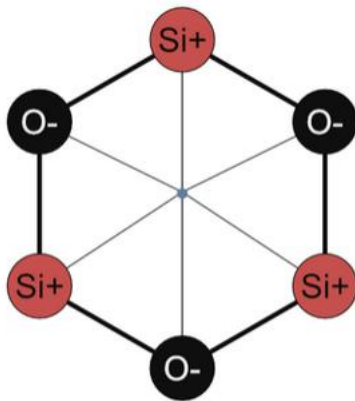
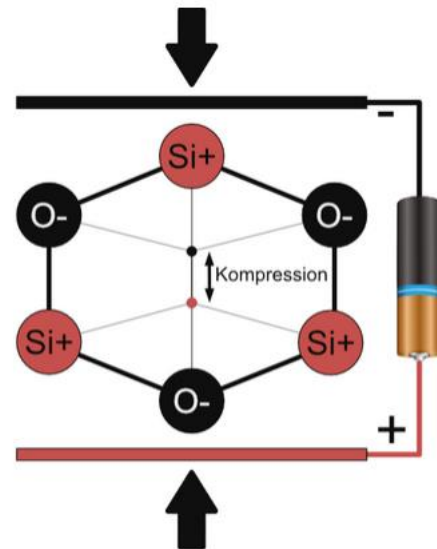
Mit dem Taster SWPower können empfangene Nachrichten nach dem Lesen „gelöscht“ werden. Das heißt, die Nachrichten werden aus der Anzeige auf der Home-Page entfernt. Ebenfalls wird durch den SWPOWER eine anstehende Alarm Meldung quittiert.

Akustische Empfangssignalisierung:

Damit Nutzer des Pagers auch auf jeden Fall bemerken, dass sie eine Nachricht empfangen haben, haben wir zusätzlich zur LED, eine akustische Signalisierung über einen Piezo-Buzzer eingebunden. Bei Empfang eines Alarms, der im Quick Send Menu vorhanden ist, wird eine kleine Melodie abgespielt.

Piezo-Buzzer:

Piezo-Buzzer oder Summer basieren auf dem physikalischen Prinzip des inversen Piezo-Effekts. Dabei werden an elektrisch neutralen Kristallen (z.B. Siliziumdioxid SiO_2) Spannungen angelegt. Diese angelegten Spannungen sorgen dafür, dass sich der Kristall verformt und so mechanische Schwingungen ausführt. Spannung und mechanische Schwingung sind dabei direkt proportional.

Bild 8.4.1: SiO_2 elektrisch neutralBild 8.4.2: SiO_2 an Spannung gelegt

Schaltet man nun die Spannung mit unterschiedlichen Frequenzen ein und aus entstehen daraus Töne.

Desktop Applikation:

Damit bei der erweiterten Bedienung des Pagers über einen Laptop oder PC keine nervtötende Terminal-Programme und Converter benötigt werden, haben wir eine Steuerung des Pagers über eine Desktop-Applikation kreiert.

Mit dieser ist es Möglich benachrichtigt zu werden wenn eine Nachricht auf dem über Bluetooth verbundenem Pager eingetroffen ist. Ebenfalls war vorgesehen das man die Quick Send Nachrichten abändern kann und z.B. den Namen des Gerätes ändern kann. Dies wurde jedoch aufgrund Zeitlichen Mangels nicht vollständig umgesetzt.

ESP32:

Der ESP32 ist das Herzstück des Pagers. Er verbindet alle Module und steuert diese an. (LoRa, Bildschirm, Taster, Buzzer, LEDs). Wie verwenden das vorgefertigte ESP32 Wrover Module. Features und Funktionalität des Moduls können in dessen Dokumentation nachgelesen werden (siehe Literaturnachweise).

LoRa-Ansteuerung:

Das LoRa Modul benötigt eine Ansteuerung über UART. Die UART-Konfiguration ist dabei, von der vorherigen LoRa-Konfiguration abhängig. Bei uns also 8N1 und eine Baud-Rate von 9600bps.

(UART Dokumentation -> Literaturnachweise)

Screen-Ansteuerung:

Der 1,3" Screen wird über IIC, mit Hilfe der „Adafruit_GFX“ und der „Adafruit_SH110X“ Library angesteuert.

Ablauf der Erstellung:

Wir starteten mit einem Testaufbau der aus einem LORA Modul und einem UART Converter bestand und einem LORA Modul mit einem ESP. Dieser Testaufbau wurde von uns am Längsten benutzt um die Grundlagen der LORA Module kennenzulernen und die Basis des Codes zu schreiben.

Nach der Fertigstellung der Grundlagen haben wir mit dem Design eines ersten Prototypen gestartet, wobei uns nach kurzem Testen gleich einige Fehler im Design aufgefallen sind, wir hatten ein sehr starkes Tasterprellen, welches durch die internen PullUps des ESPs entstanden. Wir haben anschließend noch einen Fehler begangen der uns später noch einholte, wir haben die Bauteile nicht gesocketed sondern direkt drauf gelötet.

Nach dem wir mit der ersten Version des Boards die Software des Pagers schrieben, haben wir eine zweite Version erstellt in welcher wir unsere Fehler ausglich.

Gegen Ende finalisierten wir die Software des Pagers selbst und starteten mit der Bluetooth implementation und der Windows Software

Probleme:

LORA Konfiguration

Aufgrund der schlechten Dokumentation unseres LORA Bausteines wussten wir Anfangs nicht wie wir das Board ansteuern sollen oder wie wir den Baustein konfigurieren sollen. Es dauerte einige Zeit bis wir die richtigen Einstellungen getroffen haben und über LORA Daten senden konnten -> mittlerer Zeitverlust

Screens

Aufgrund des mehrfachen Wechsels der Boards sind uns die Bildschirme eingegangen, wir probierten diese wieder und wieder zu reparieren, doch einer bei einer der beiden Screens war das Backlight defekt, welches wir nicht wieder reparieren konnten.

Bluetooth in WinUI3

Gegen Ende des Projektes als wir mit der Windows App starteten, hatten wir massive Probleme, da es uns Anfangs nicht möglich war die Bluetooth API von Windows korrekt anzusteuern, doch als wir ein externes NuGet nahmen, statt der Windows direkten API, funktionierte der Aufbau der Verbindung, jedoch dauert sie ein wenig länger.

Bedienungsanleitung:

Eine Ausgiebige **Bedienungsanleitung**, inklusive einer **Erklärung** der Implementation ist unter <https://github.com/basti-debug/pager> zu finden **Code, Software sind ebenfalls in der Github Repository zu finden**

Literaturnachweise:

Piezo-Buzzer

<https://edistechlab.com/piezo-summer-buzzer/?v=fa868488740a>

ESP32 Wrover

https://cdn-shop.adafruit.com/product-files/3384/esp32-wrover_datasheet_en.pdf

UART

https://www.ti.com/lit/ug/sprugp1/sprugp1.pdf?ts=1686988071833&ref_url=https%253A%252F%252Fwww.google.gr%252F