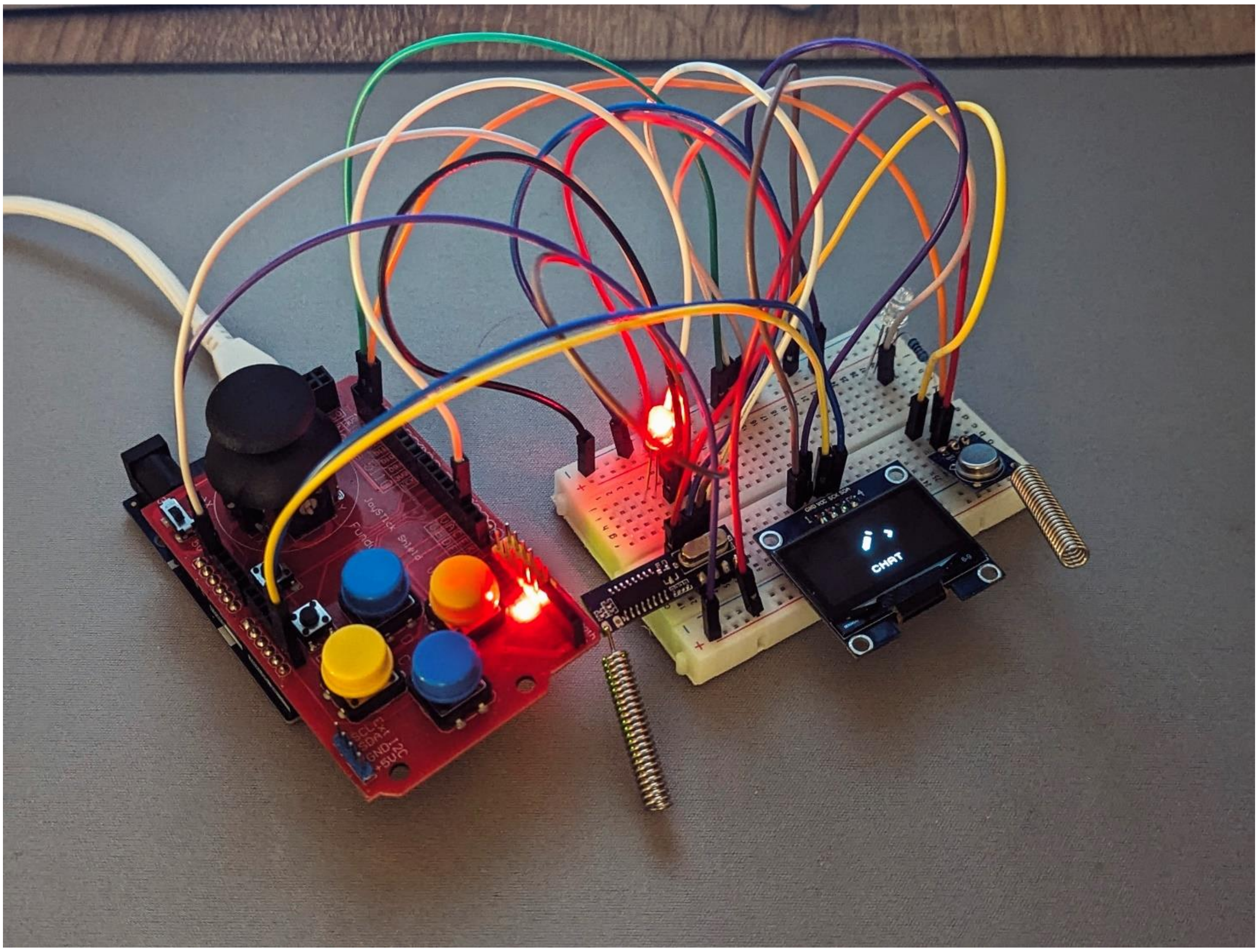
RF MESSENGER

Using Arduino Uno Rev3 and NiceRF 433MHz transmitter and receiver

https://github.com/tuvumba/RFMessenger



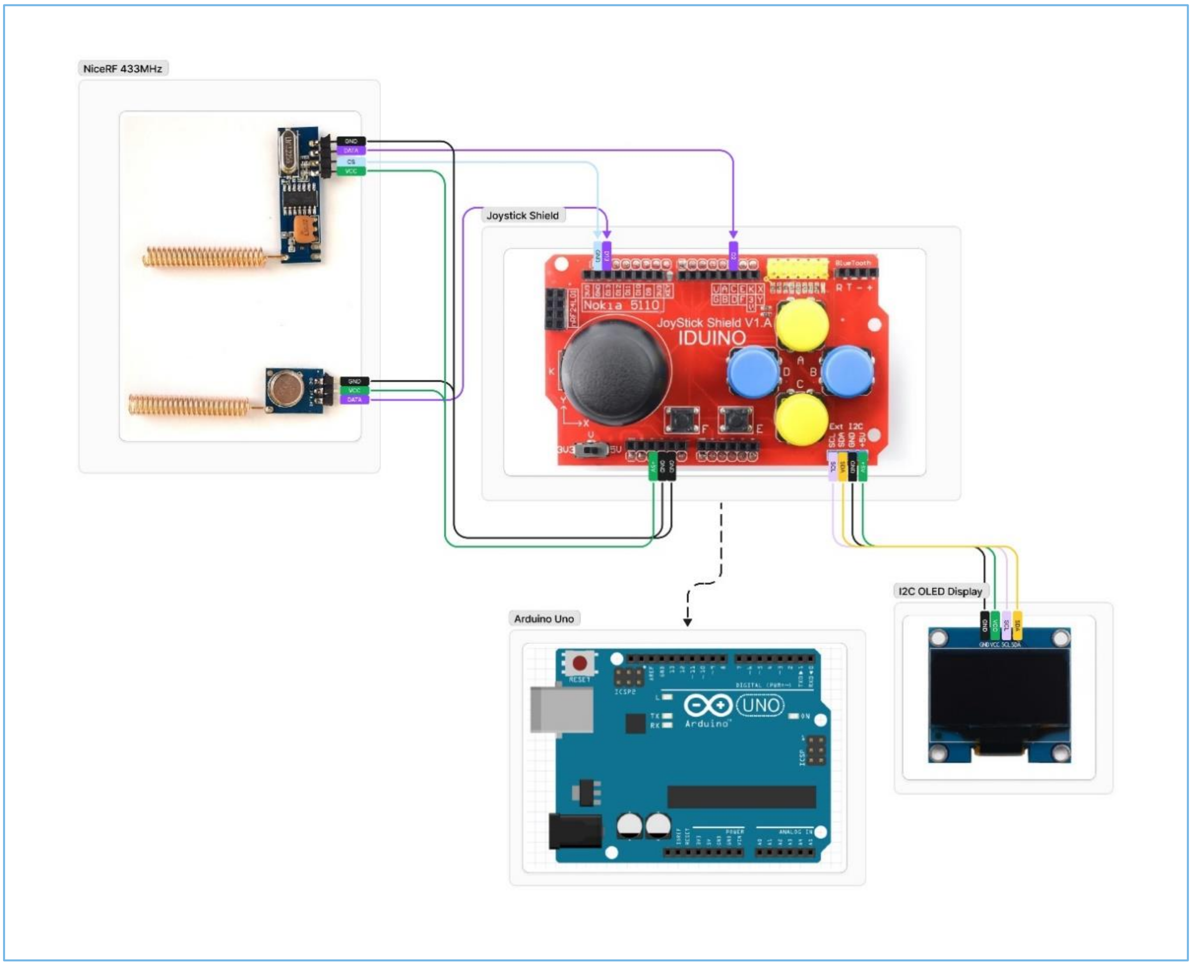
Key features:

- Minimalistic messenger-like UI
- Transmission up to 600 meters (well, should have been)
- Ease of use – only joystick and 4 buttons required.

Required hardware:

- Arduino Uno Rev3
- I^{2}C 1.3" OLED Display (SH1106)
- NiceRF 433MHz (STX882 transmitter, SRX887 reciever)
- Arduino Gamepad Shield

ASSEMBLING THE HARDWARE



Above is the connection scheme for this project.

1. Connect the Joystick Shield to the Arduino.
2. Connect the parts according to the diagram.

Assembly tips:

- You may want to use the breadboard for connecting the GND and VCC pins on RF modules.
- RF-transmitter’s DATA pin may be connected to any free D pin on the Arduino, in my implementation it is connected to the D13 pin.
- RF-receiver’s DATA pin **must** be connected to D2, as it is one of the pins supporting interruptions on Arduino Uno.
- Be sure to check that connection to the display is adequate as it may affect the image quality.
- Joystick Shield mode **must** be set to 5V.

USER MANUAL

QUICK START

- Connect the hardware according to the scheme in the "Assembling the hardware" section.
- Install the following libraries:
  - ➔ U8g2lib https://github.com/olikraus/u8g2
    Manages the display.
  - ➔ Array https://www.arduino.cc/reference/en/libraries/array/
    Used in the implementation for chat storage.
- Compile and upload the sketch to your Arduino board. https://docs.arduino.cc/software/ide-v2/tutorials/getting-started/ide-v2-uploading-a-sketch/

- Repeat the process for other participants.

USING THE RF MESSENGER

- Move the Joystick left-and-right to access different menu entries.
- Confirm your choice using the DOWN button (marked C on the Joystick Shield).
- Go back using the RIGHT button (marked B on the Joystick Shield).
- When in CHAT, press the DOWN button to open the on-screen keyboard.
- The RF Messenger stores up to 32 last messaged. Scroll through them by moving the Joystick UP or DOWN in the CHAT menu.
- Additional advanced options are accessible in the DEBUG menu.

PROJECT SUMMARY AND AFTERWORD

During development of this project for BI-ARD at CTU FIT, I have managed to build a potentially working radio messenger with minimalistic UI and modern controls.

Shortcomings of this project:

- I couldn’t establish a radio connection between Arduinos.
  I have tried different boards, different settings but still nothing.
  Most probably, the unit I have bought is **broken**.
  But I truly believe that this project might work especially well when equipped with better RF modules (LoRa, for example).
  However, the only things that would need to be changed to adapt the project to other modules are the functions **sendText**() and **refreshMessages**() in the **Menu** class.

Key takeaways and notable things:

- The screen is refreshed only when needed, this helps with the image quality as there is no flickering. (especially the keyboard took some time to figure out)
- I have learned how to solder! No, really. I made a dumb decision when buying the modules without the pins already soldered to the board and had to look for a solution.
- I have learned the differences in the architecture of R4 and previous generations the hard way. Nonetheless, I am planning to continue developing this project, maybe even adapting the RCSwitch library to the newer generation of Arduinos.