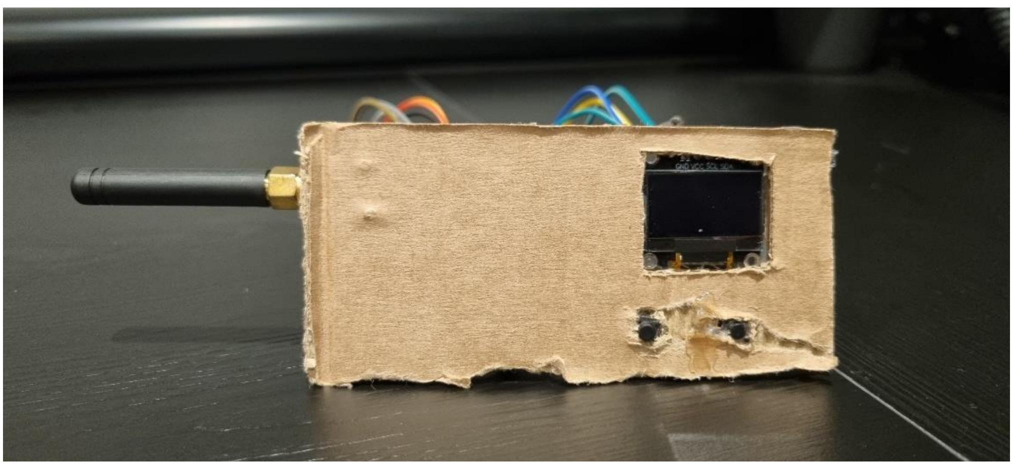
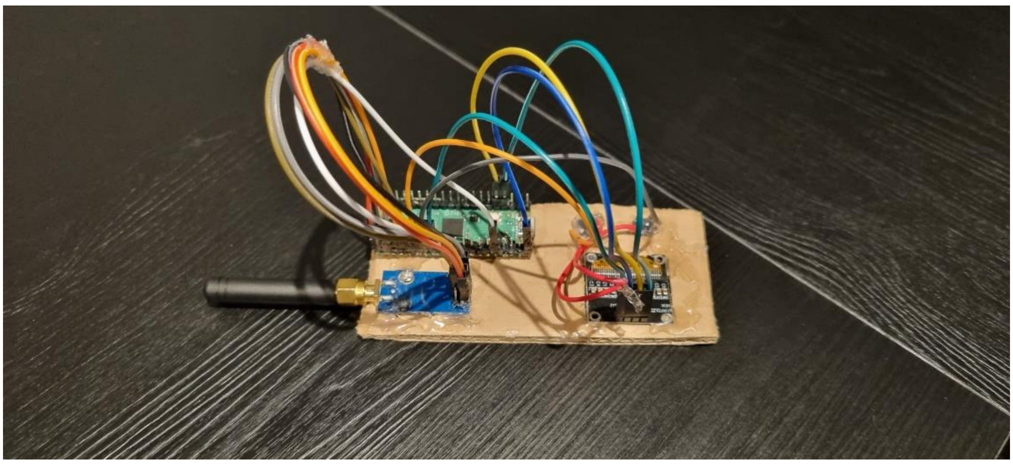
September 2025

In just two weeks, I built my first project that fully integrates software with hardware, powered by MicroPython.

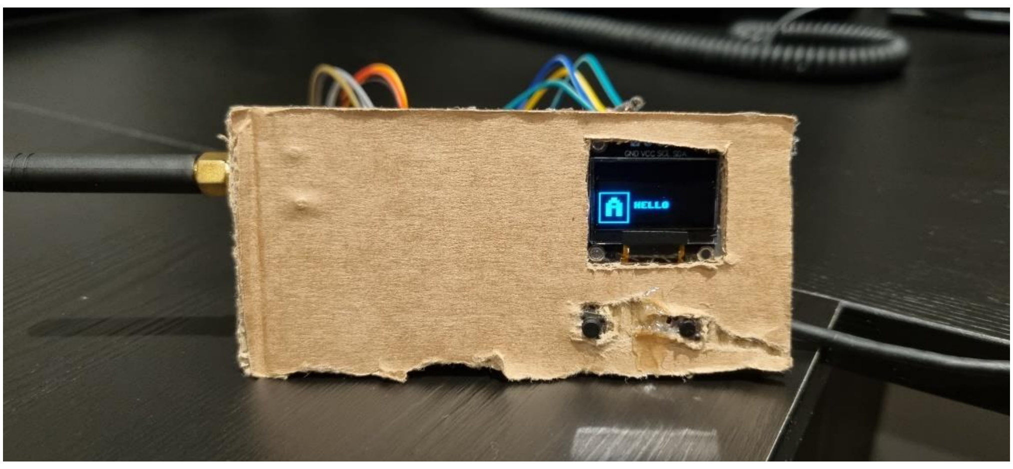
Each one of the Raspberry Pi Texting modules was assembled using a Raspberry Pi, display, radio module, two buttons and cardboard.

Below is an example of one of the unpowered Raspberry Pi Texting modules.

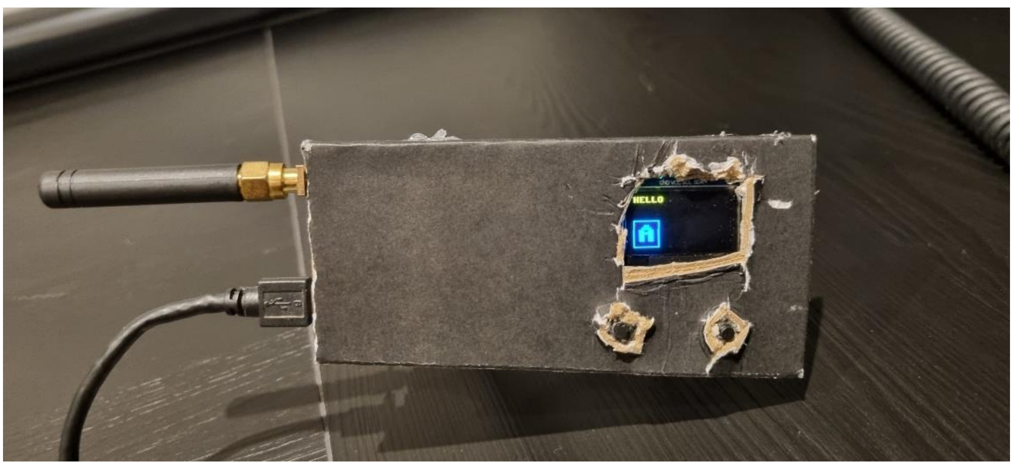




The module works when powered on by allowing the user to scroll through a list of characters when the left button is pressed, when the user has found the desired character they can press the right button to amend the character to the message line or alternately send the message. The character list and message line displaying selected characters can be seen below.



When the message has been sent it will be received by the other Raspberry Pi Texting module and displayed in an alternate display segment than the message line to avoid interference. The message received can be seen below.



I chose the Raspberry Pi Pico for this project because it supports MicroPython and provides easily accessible memory for modifying code. An SSD1306 OLED display was used due to its straightforward pixel control making it ideal for displaying characters. For communication I selected a CC1101 radio transceiver to maximize texting range and minimize interference from other signals such as Wi-Fi, which operates at 2.4–5 GHz, while my module runs at 433 MHz. Two buttons were included to simplify user input and navigation. Finally, the components were mounted on a cardboard structure, though I plan to design and 3D-print a more robust enclosure in the future.

Overall, the code for this project was 284 lines long and can be used without change if more than two Raspberry Pi Texting modules were added to the network.

Made by Oliver Ron Andreasen.