

G22 Install/Operation Instructions

Please note that the project is not yet fully ready for deployment onto the intended target environments, nor for any kind of public use.

As such, these instructions do not reflect the intended final functionality of the project.

They should however allow you to install the needed pieces of software to demonstrate the functionality as it was at the May 2nd deadline.

User Interface Installation

1. Download the “ChargerArcade” folder from the GitHub repository to a Windows computer (to any directory you want)
2. Open the .pro file within the folder using the latest version of Qt Creator (v6.7.3 or newer, though older versions may also work, but are not guaranteed)
3. It will likely ask you to/walk you through configuring some parts of the .pro file and/or related prompts, just confirming with the default values should work, given that the Qt installation is not incomplete
4. Create a file in the computer’s C:/ directory called “CharcadeID.txt”
5. Make the first line of this file be the ID of the terminal (in digit form, no letters) (count should start from 1)
6. Make the second line be the path to the “ChargerArcade” folder that was previously downloaded (use / instead of \, and this path should start with “C:/” and end with “ChargerArcade/”)
7. Run the program using the green arrow in the bottom left of the Qt Creator editor

Package Editor Installation

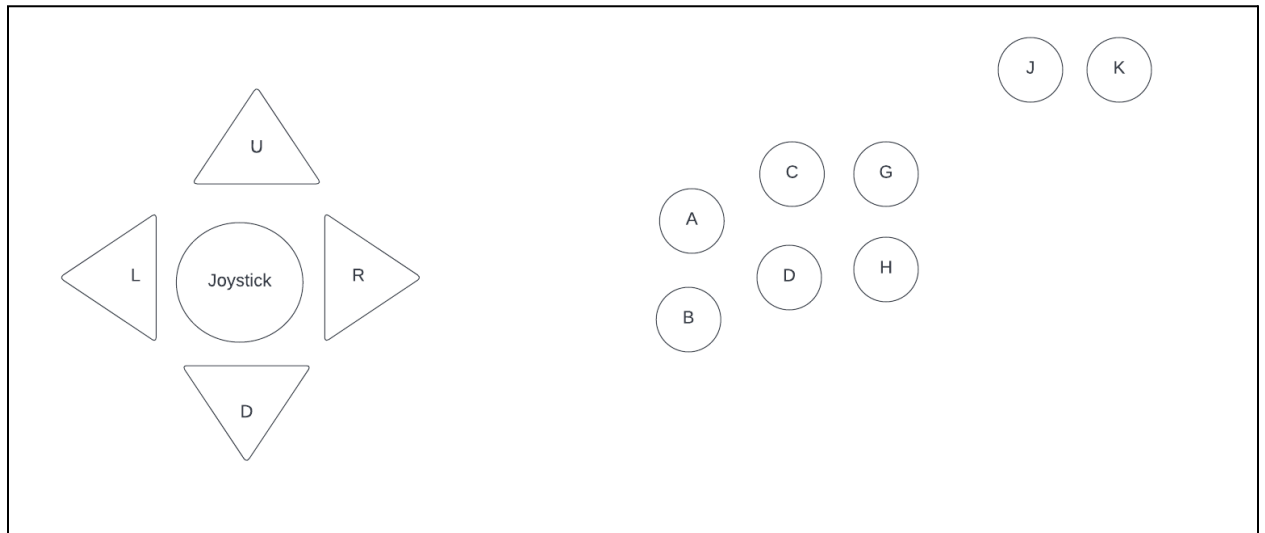
This software can be installed and run similarly to the UI (which is detailed above), without steps 4-6.

Hardware Setup

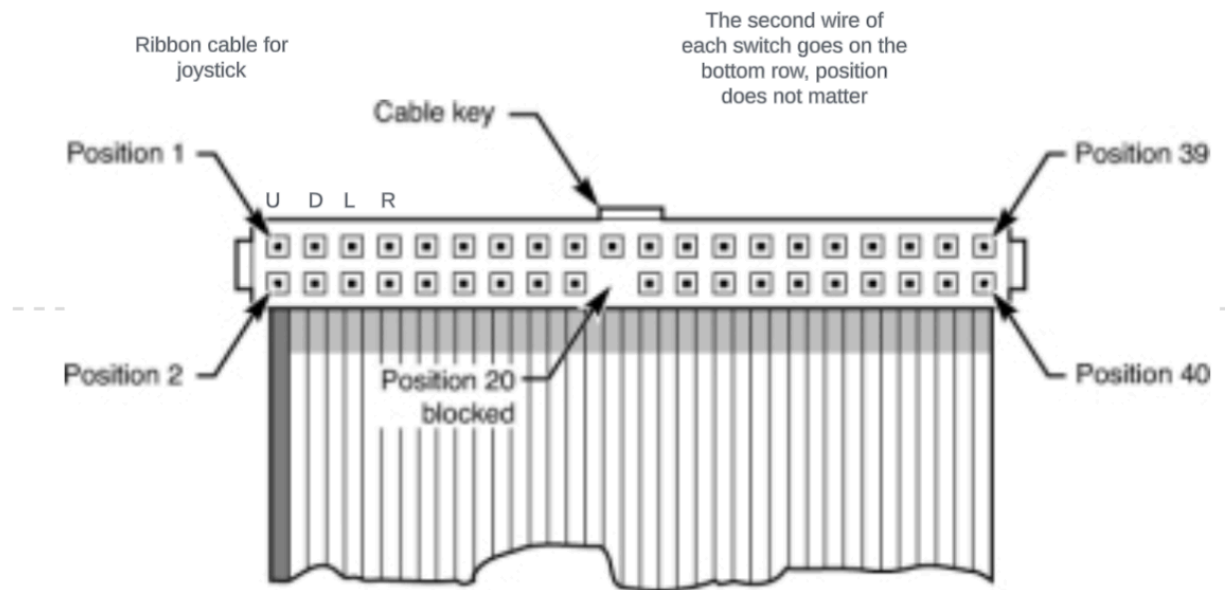
1. Wire breadboard, buttons, and joystick to the Raspberry Pi Pico according to the below pictures
2. Each button in the below pictures has an input and an output wire. The output wire is the wire that goes to the Pico, and the input wire goes to the red power rail of the breadboard.
3. Similarly, the joystick has an output and input wire for each direction, where the output wire goes to the Pico, and the input goes to the power rail of the breadboard.



4.

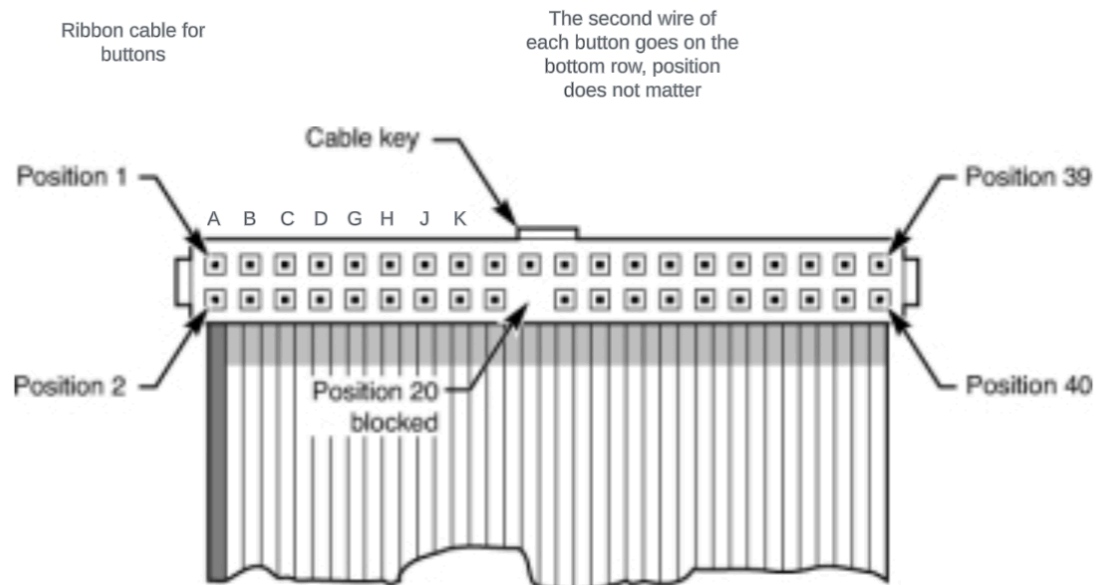


5. From the controls panel, wire each letter to one of two ribbon cables:
 - a. One ribbon cable for the joystick



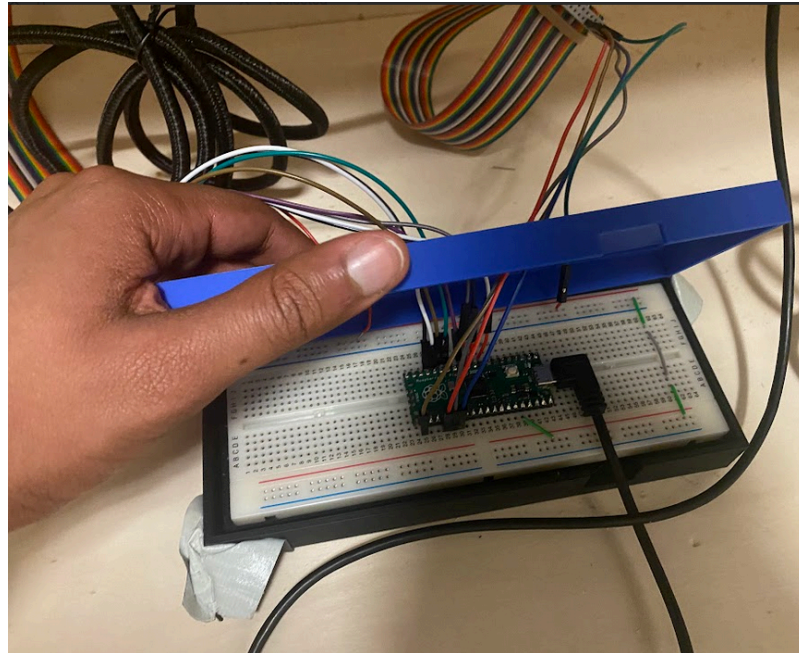
b.

c. One ribbon cable for the buttons



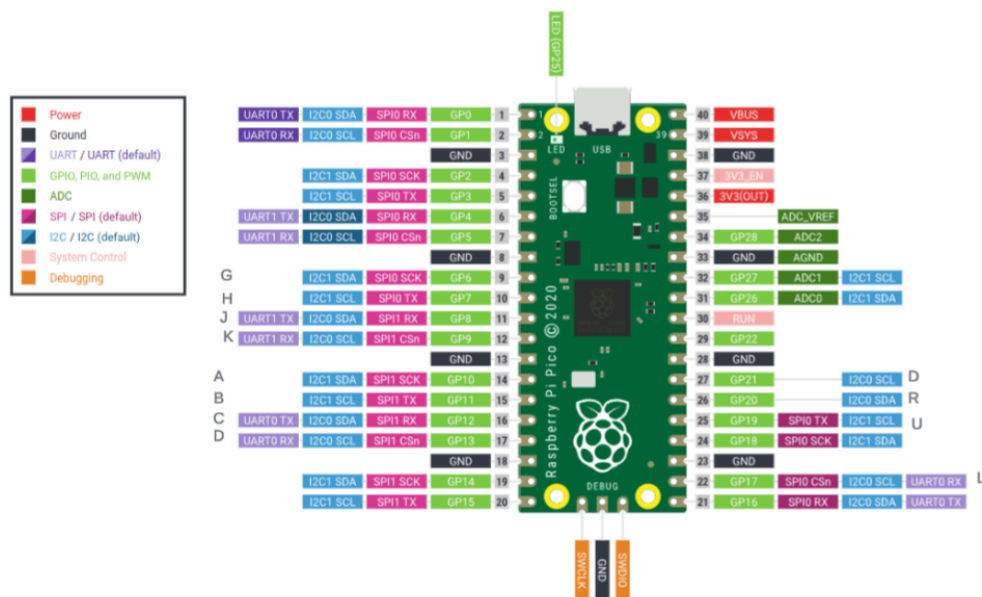
d.

- e. Power wires for the joystick and buttons go on the bottom row of each ribbon cable, in no particular order.
- i. From each ribbon cable, wire the letter outputs of each button into the Raspberry Pi Pico as shown below



6.

Raspberry Pi Pico Pinout



7.

- All power outputs of the ribbon cable go to the red power lines of the breadboard, which is wired to the 3V3(OUT) pin of the Pico to get 3.3V output from.

- Install Arduino: <https://www.arduino.cc/en/software>

- Open Arduino to a blank file

- Navigate to the serial folder in the directory and download testament.ino:

https://github.com/greenheron02/project_repo/blob/main/serial/testament.ino

12. Follow the instructions in the link to upload with the Pico:
<https://www.upesy.com/blogs/tutorials/install-raspberry-pi-pico-on-arduino-ide-software>
 13. Open the testament.ino file using Arduino
 14. Upload the file to the Raspberry Pi Pico
- Arcade Input Receiving Setup
15. Install Python via this install link
<https://www.python.org/downloads/release/python-31016/>
 16. Open IDLE via the start menu bar
 17. In IDLE run the following commands: import sys, sys. executable
 18. Copy the link and open the terminal
 19. Paste the link into the terminal, followed by pip -m install pyserial
 20. If it doesn't work, look for python.exe in your folders and use that path instead of the copied one
 21. Navigate to the project repository:
https://github.com/greenheron02/project_repo/tree/main
 22. Download the p2p2p2p.py, serializing.py, and keyboardinputting.py files
 23. In the Windows start menu, search up IDLE and open it
 24. After IDLE is open,
 25. Open the files using IDLE, under the File tab
 26. Ensure they are saved to the same folder, otherwise move them to the same folder
 27. Run the p2p2p2p.py file
 28. You now have input, if the Raspberry Pi Pico has been uploaded to correctly

UI Operation

setup

1. For the first time a terminal is set up (or a new one is added to the network), the following variables need to be changed within the Charcade.h file:
 - "TerminalCount": the total number of terminals in the network
 - "addresses": get the IP addresses (for the network) of each terminal (which can be set through Windows settings, look it up if you would like to do so), and set the "addresses" vector to contain each one, in the style that is already shown, listed in the order (also put a colon and a number after them, just make sure this number is a valid port) The "addresses" vector MUST be the same on each terminal.
 - "IdReference":
2. Run the program

3. Follow the instructions from part 20 of the Hardware Setup to run the arcade inputs to keyboard inputs script 'p2p2p2p.py'
4. After running the script, you will start to receive keyboard inputs from the hardware

User operation of UI Software

The user interface is designed to be operated by the user using nothing more than the joystick and the green and red buttons. These operate as follows:

- On the title screen, press the green button to enter into the selection screen
- On this screen, you can move the cursor with the joystick and enter a per-game sub-menu with the green button
 - In this menu, there are options listed that you can navigate through with the joystick and select with the green button.
 - Note: Using the invite others option will only allow them to launch a specific pre-installed game. This, like all other issues detailed in these instructions, will be fixed soon, but at the current cut-off deadline, they are not functional.
- You can also enter a menu by selecting a button in the top-left of the screen. None of the options in here do anything at the moment.
- When a game is started (using the sub-menu), the game will then begin
 - NOTE: Attempting to launch a game that is pre-installed might lead to errors as they have not been equipped with complete information.
- You cannot leave a game using the controls just yet at this stage of the project. To leave just use typical Windows methods of leaving an app (alt-f4, switching windows, etc.). Then press "E" on the keyboard while focused on the ChargerArcade editor to return back to the selection screen and also restore the UI control functionality back to the controls.
- While in the selection screen, you can enter a USB drive into a USB slot on the system. The program will scan it for package files, which can be generated with the Package Editor software, and will load the games into the selection screen.

- Unplugging the USB drive will trigger the added games to promptly disappear from the software
- Regardless of any per-game controls setups, pressing the right button in the top right of the panel while in a game will replace all controls functionality with a “mouse mode”, where the joystick can control the mouse and the green button can click the mouse. Pressing the right button again will make the controls return to normal.
- Pressing the left button in the top-right corner will have the address “192.168.0.3” be typed into the program. This was intended to be the address of the host system, but it does not as of right now dynamically change with the terminals or addresses.
- If a USB drive is plugged in that has a file called “key.txt” on the root directory of the drive, with the text “HelloIAmKey” as its contents, the UI will recognize this as a Root User accessing the system, and it will flip into “edit” mode.
 - Edit mode currently does nothing just yet

Operation of Package Editor Software

- Most of the software’s operation should be either intuitive, or explained through the built-in help messages.
- Note: Do not save files directly to a USB drive, as this seems to not work as of right now. If you wish to load the package onto a USB drive, simply save it to a spot on the computer, and then move it to the USB drive.