

## Introduction

The clock uses six Waveshare 2-inch 240×320 LCD Display Modules each with their own ST7789 driver. They are about \$15 each. The clock also uses a Raspberry Pi 4B single board computer which is about \$60. Additionally, there is a solderless bread board, a 40-pin ribbon cable and various jumper wires all of which come to about \$30. The RPi talks to the LCDs over an SPI interface – the wiring for this is provided later in this document.

When you first get a Raspberry Pi, you must sort of set it up, easy – once you’ve done it. The first step is to plug in a USB mouse and a USB keyboard via direct connection or wireless. Like on a new PC, when you first fire it up you need to set up a username and a password. Also, for the clock, you’ll need to install the Python Interpreter and enable the SPI interface and then, finally, connect your RPi to your Wireless LAN.

For a fully built clock most this will already have been done (all except for connecting to the LAN) and once it’s done you can get rid of the RPi mouse/keyboard and use your PC to connect to the RPi. You’ll only have a command line interface to the RPi, but this is really all you need.

## Handy Commands

Assuming your RPi shows up on your LAN at IP 12.34.56.78, then you can connect to it via this command:

```
ssh pi@192.168.1.120
```

The PC terminal window that this command was entered in now becomes an RPi terminal window.

Now say you have some python files on your PC that you want to copy over to the RPi. To do this open a PC terminal window (you don’t need to have a ssh window open as discussed above) and enter this command:

```
scp *.py pi@12.34.56.78:~/python/spiClock
```

This will copy all files ending in .py in the current PC directory to the RPi in directory python/spiClock. This copying of files will have already been done on a fully built clock.

And for completeness, to copy up from the RPi to the PC do this:

```
scp pi@12.34.56.78:~/python/spiClock/digitScreenStyles/*.pickle ./. 
```

This will copy all files ending in .pickle from the /spiClock/digitScreenStyles directory on the RPi into the current directory on the PC.

This is handy because all the software and font styles are pushed from a PC to the following Github account: <https://github.com/sgarrow/spiClock>. Since the software is written on a PC, it’s already present on the PC, but the font styles are built (by the software previously pushed to and running on the RPi) on the RPi. So before pushing to Github the font styles need to first be uploaded from the RPi the PC.

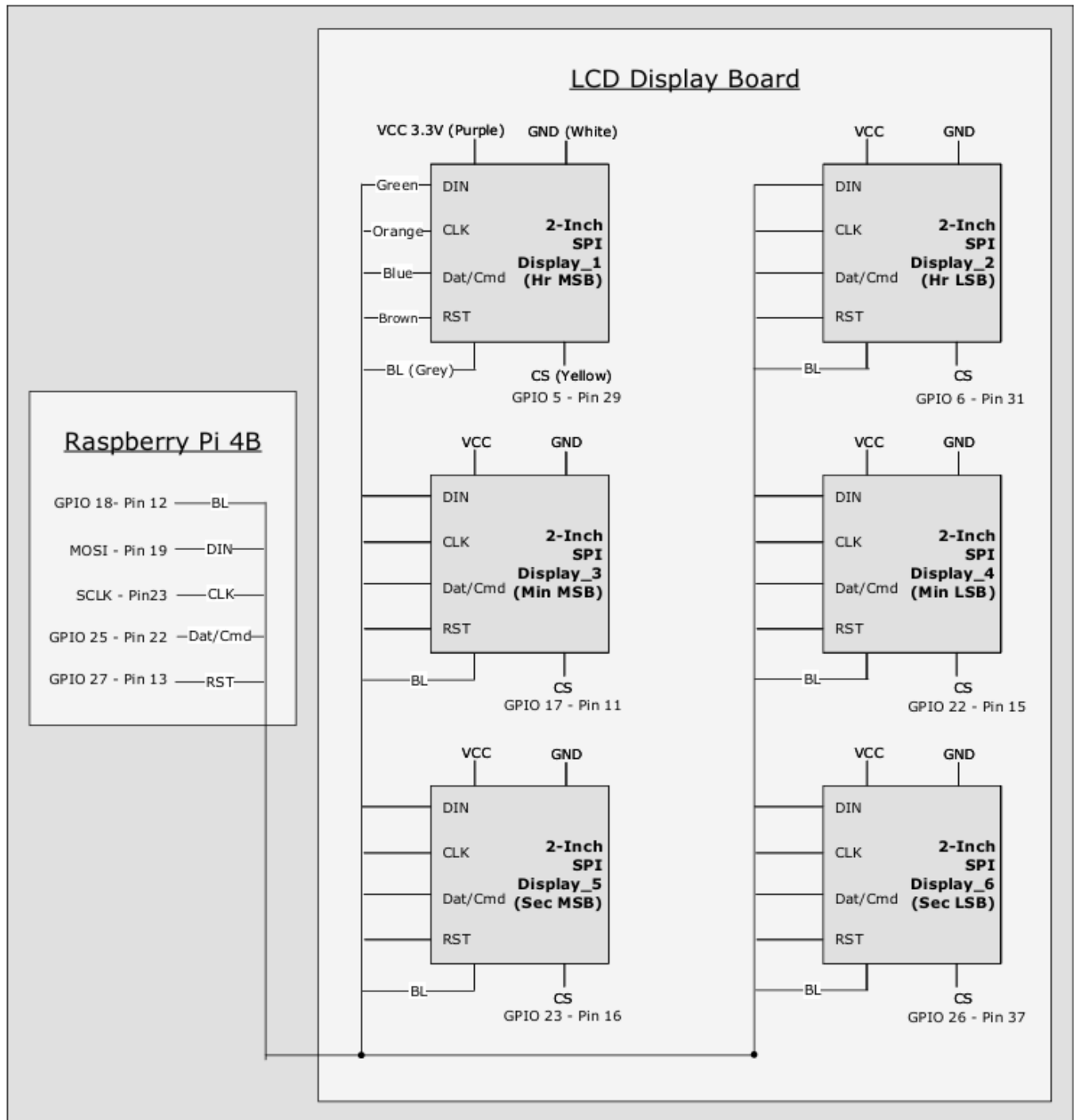
Really, for a previously built and configured clock a new user will only need to do two things: (1) – Install the Python Interpreter on their PC and then (2) copy a single file (client.py) from Github. Then to control the clock enter the following command in a PC terminal window: `python client.py`.

There’s one more thing they’ll need to do: modify one line of code to add their RPi’s LAN address and (optionally) their router’s IP address (and configure port forwarding).

Change:	<code>connectDict = { 's' : 'localhost',</code>	<code>'l' : '0.0.0.0',</code>	<code>'i' : '00.00.00. 00'}</code>
To:	<code>connectDict = { 's' : 'localhost',</code>	<code>'l' : '12.34.56.78',</code>	<code>'i' : '00.00.00. 00'}</code>

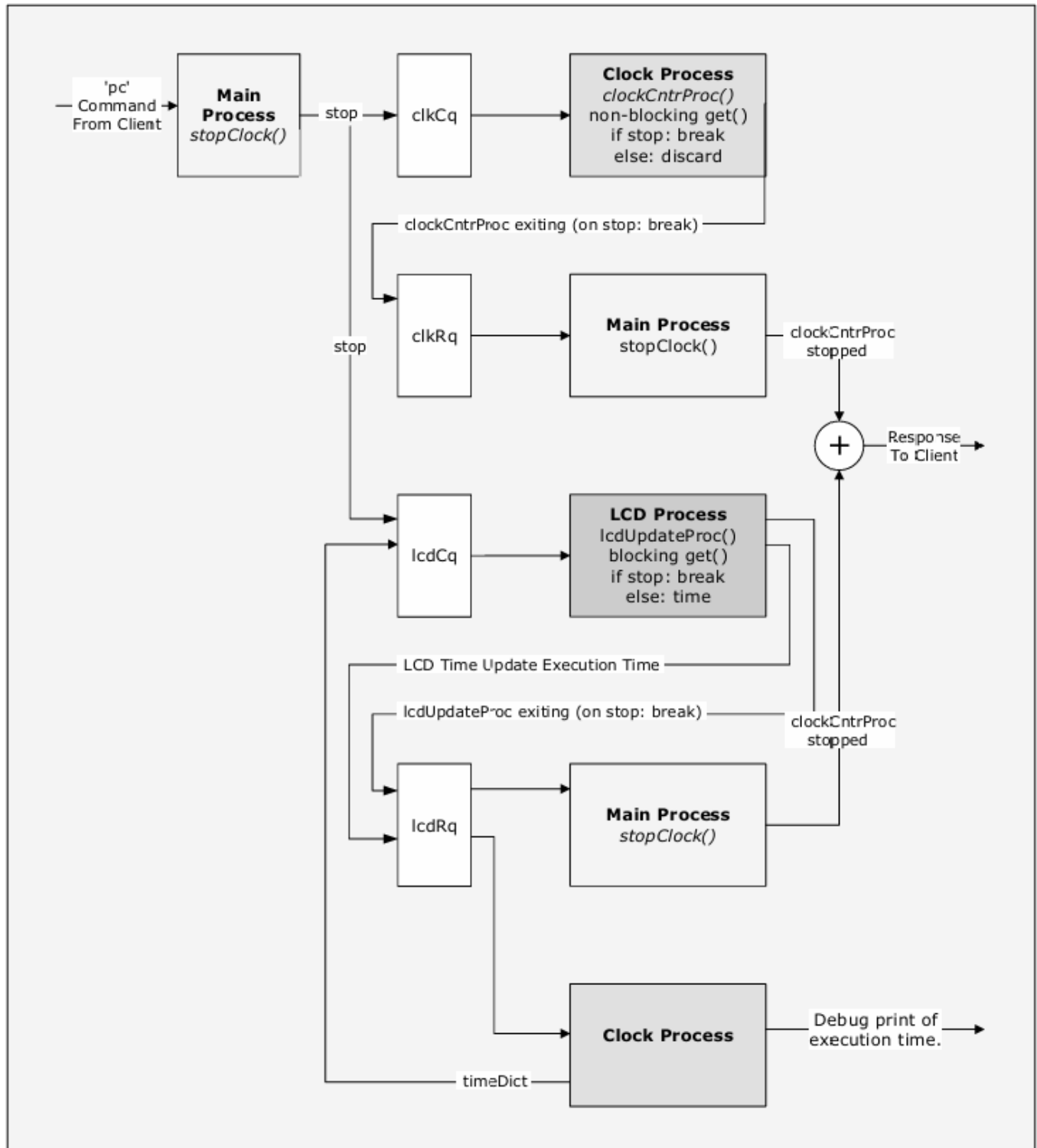
## SPI Wiring

A simplified SPI wiring diagram is presented below.



## Communication Queues

The clock runs on three separate cores. One core runs the server (Main Process), another runs the clock counter (Clock Process) and the third controls the displays (LCD Process). These three processes communicate with each other using four of Python's awesome multiprocessing-communication-queues. Two queues are used for sending commands and the other two are used for receiving responses. A simplified communication diagram is presented below.



## Running the client

Below is an example client session. After the client is started and a connection is accepted by the server (the clock) a prompt is presented. When 'm' is entered at the prompt a list of available commands is presented. When the gAs is entered a list of available font styles is presented.

```
PS C:\01-home\14-python\spiClock>
PS C:\01-home\14-python\spiClock> python .\client.py

Exception importing readline. ok to continue.

sndBufSize 65536
rcvBufSize 65536

Accepted connection from: ('192.168.1.110', 51297)

Choice (m=menu, q=quit) -> m

sc - Start Clock
pc - stoP Clock
tm - Test Menu
rh - Reset LCD HW
rs - Reset LCD SW
sb - Set Backlight
gas - Get Active Style
sas - Set Active Style
gAs - Get ALL Styles
mus - Make User Style
lc - List Commands
ks - Kill Server

Choice (m=menu, q=quit) -> gAs

turquoiseOnOrange
whiteOnBlack
blackOnWhite
orangeOnTurquoise

Choice (m=menu, q=quit) -> |
```