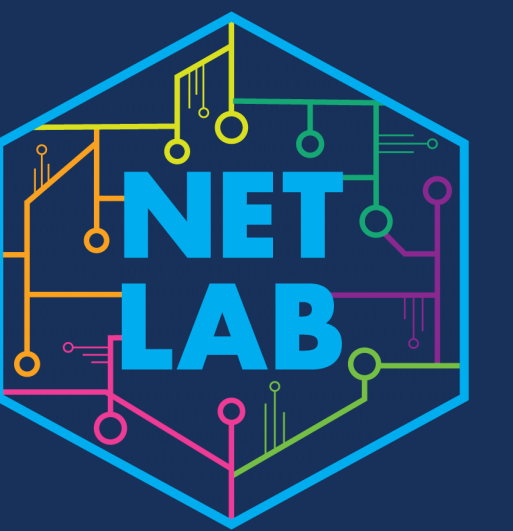




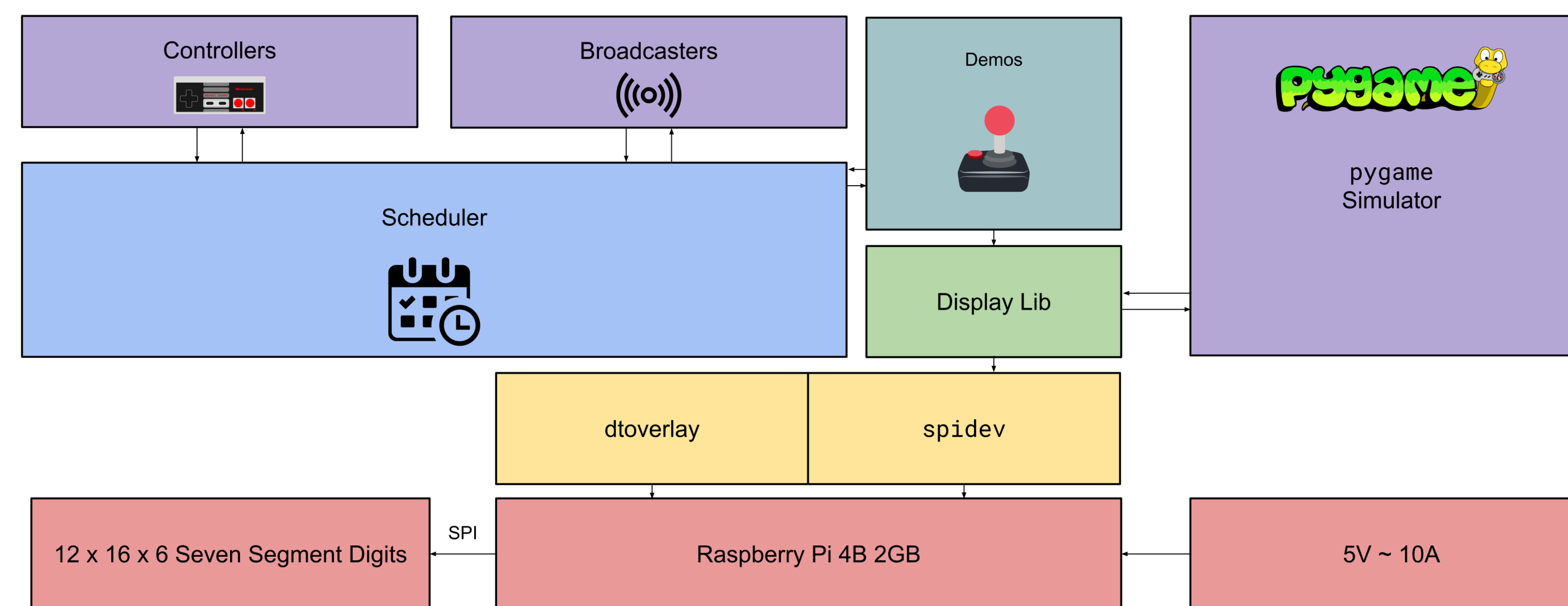
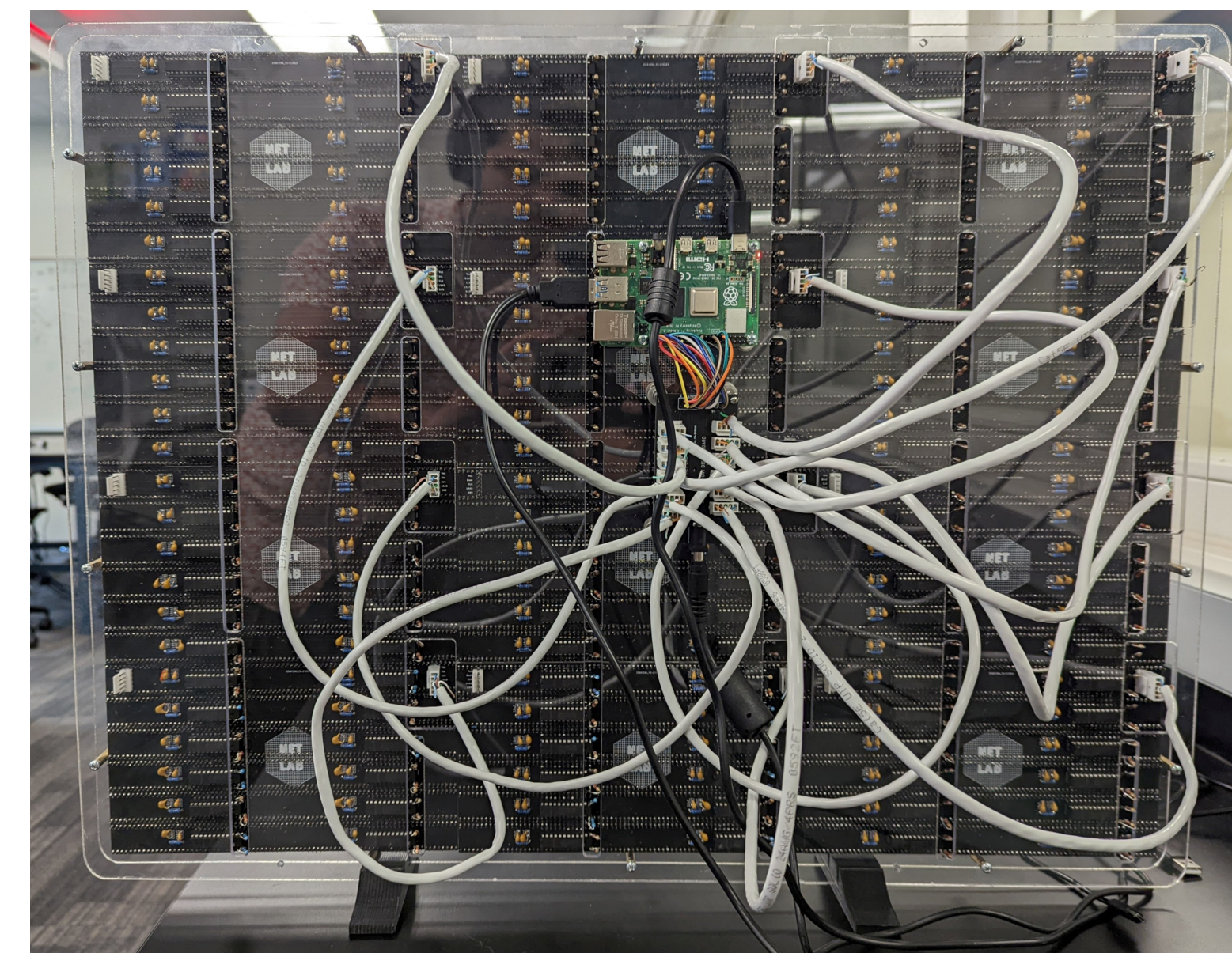
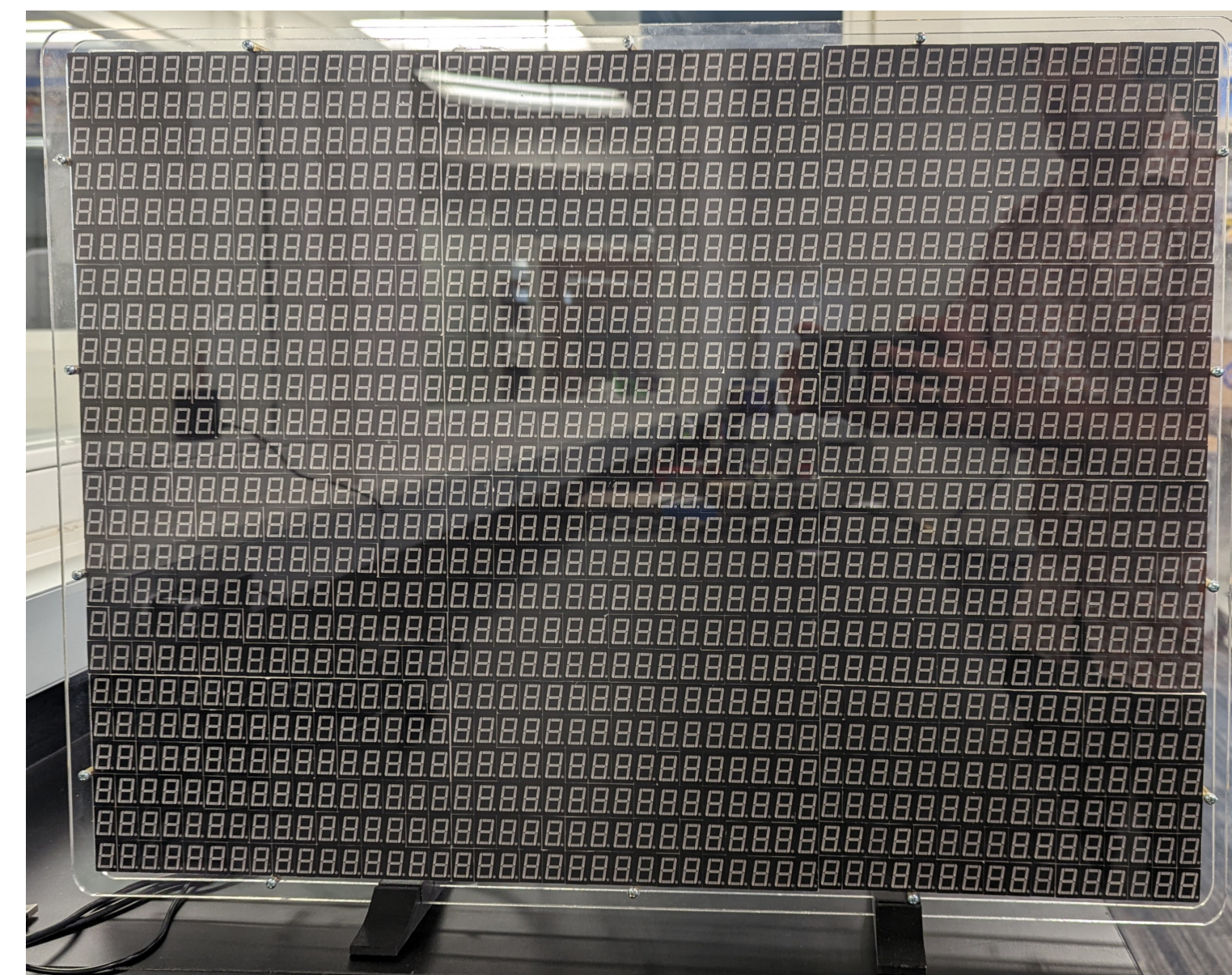
SSS: Building a Seven Segment Sign

Chris Kitras, Philip Lundrigan, Ashton Palacios
Brigham Young University



Introduction

Using a Raspberry Pi, **1152 seven-segment displays**, and Python, we built a Seven Segment Sign (SSS). The project is designed to make something familiar and boring (seven-segment displays) into **something remarkable**. We created a **Python library** to abstract all of the hardware details of the sign to make it **easy for new Python developers** to write applications for the SSS. We have created **multiple games and demos** to show what the SSS can do. And, of course, we even **ported Doom**. We make extensive use of Python generators to design our system. We made our system modular so that it can **receive input from multiple sources**, like a game controller or the network using a smartphone.



Architecture

Scheduler:

Generators are the core underpinnings of our system. The scheduler calls each subsystem's run/tick function which quickly performs a single action and yields control back to the scheduler.

Controllers:

Demos can have input ranging from a physical game controller to a web based app.

Broadcasters:

Demo and system outputs can be broadcast over the internet to different services.

Display Library:

Custom Python library that abstracts away drawing to a physical screen or simulated pygame screen.

Demos

Frame Rate:

Parameter to inform scheduler how many frames a second to generate.

Run Time:

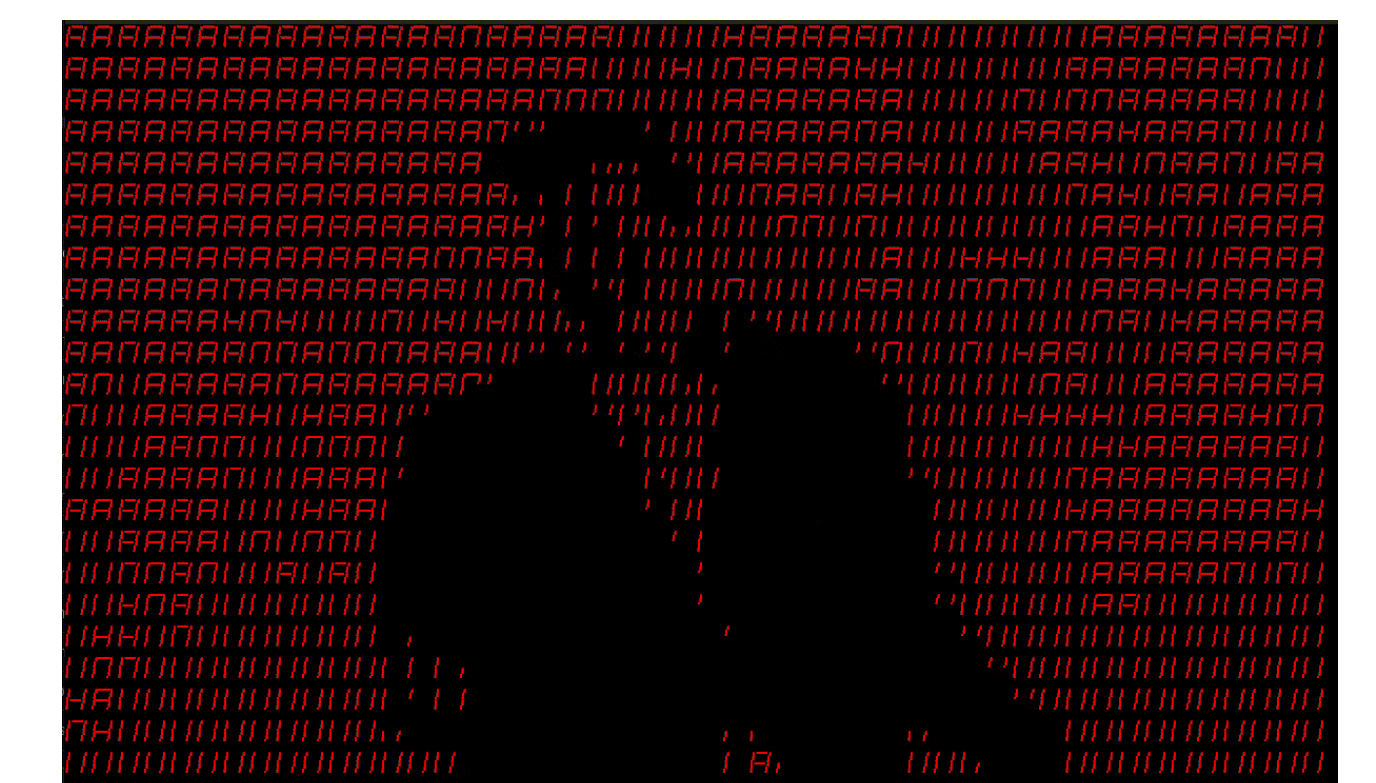
Parameter to inform scheduler how long this demo should be run.

Input/Output:

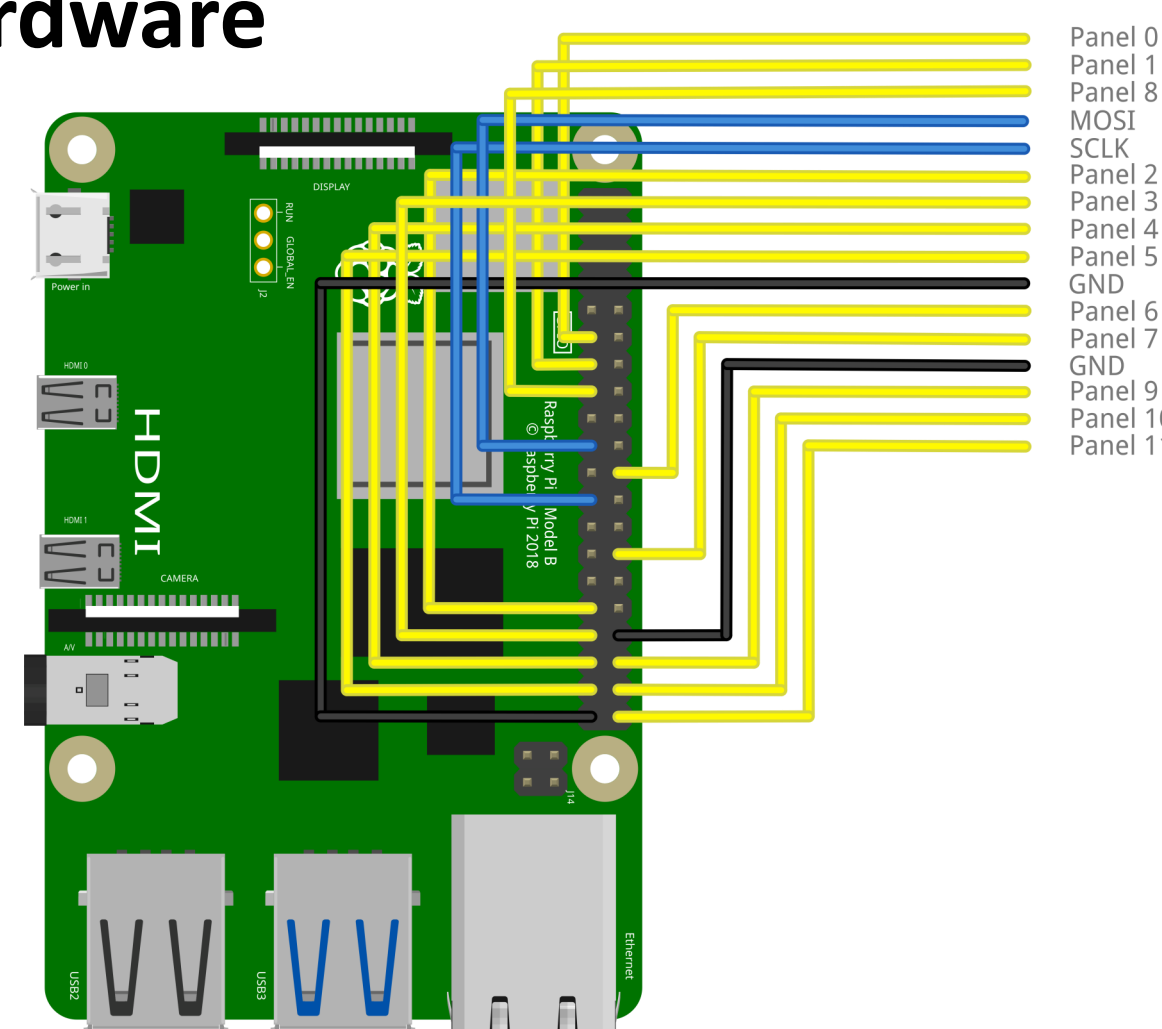
Queues that allow controller/broadcast data communication to and from the demo to the scheduler

Run:

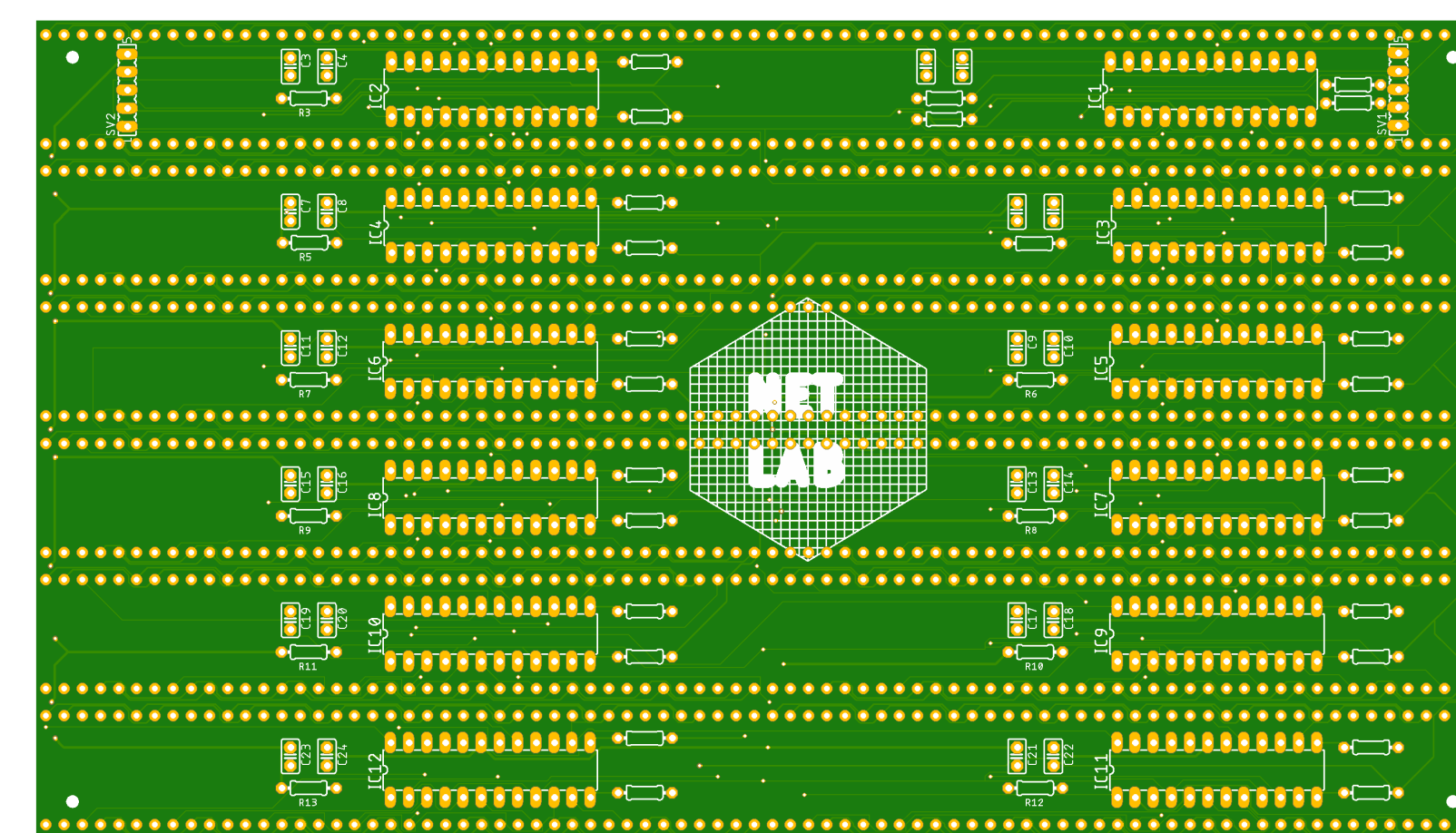
Generator function that controls demo logic that is called by the scheduler.



Hardware



- Panel 0
- Panel 1
- Panel 8
- MOSI
- SCLK
- Panel 2
- Panel 3
- Panel 4
- Panel 5
- GND
- Panel 6
- Panel 7
- GND
- Panel 9
- Panel 10
- Panel 11



Read the Docs



GitHub

