CRESTRON Masters 2021

MASTERS 2021

The premier event for the AV industry's top programmers and designers

MCP-101 **Fundamentals of C# for Crestron**

Instructor: Tim Gray


CRESTRON
1
Like what you're seeing and learning at Masters this week?
2
Tweet your photos, videos, and quotes using #CrestronMasters
Product Development

Session General Information

- **This class will be recorded.**
- **The video & the PowerPoint will be posted to Crestron Online Help ID 2015.**

Crestron Masters presentations and videos

- **Please use the Teams Chanel for this class to submit questions about the presentation.**
- **Note all questions may not be answered during the session due to the number of attendees.**
- **All questions with answers will be e-mailed and posted to OLH ID 2015**

- **For concerns about registration please e-mail RSVP@crestron.com**

MCP-101 **Fundamentals of C# for Crestron**

Instructor: Tim Gray

Getting Started

Crestron
VIRTUAL CONTROL

- Offers a centralized server-based alternative to individual hardware-based control systems in every room
- Provides a virtual control system for each room over a network

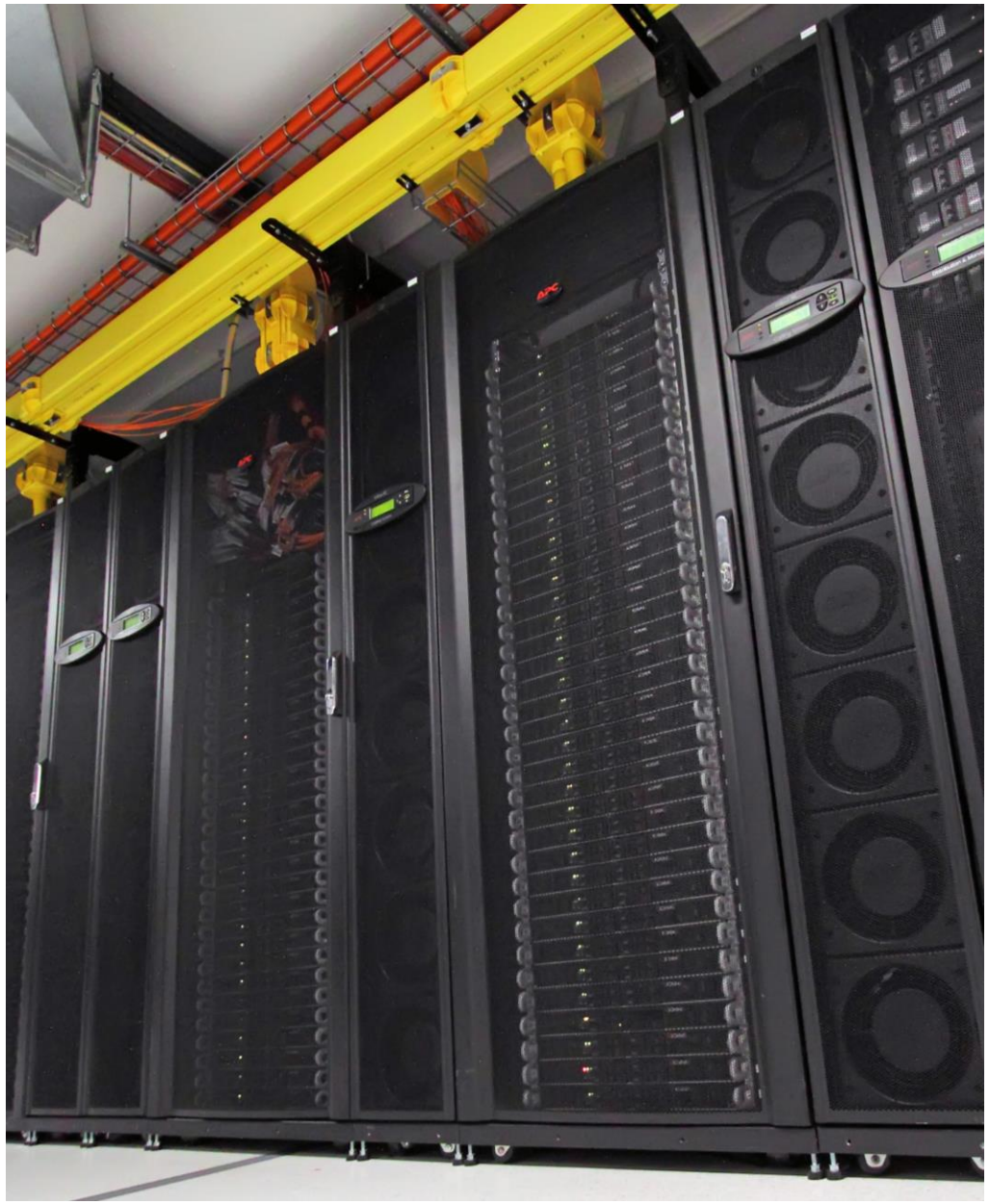


Image Courtesy of CSIRO

Files for this class are available on GitHub

https://github.com/CTI-Tim/Masters2021-MCP-101

Or Search Github for….

Masters2021

or

Masters2021-MCP-101

Your Developer Environment

- Visual Studio 2019 Community
  - Class Library (.NET Framework 4.7)
- Crestron Libraries using NuGet
  - Program
  - Program Library
  - Library

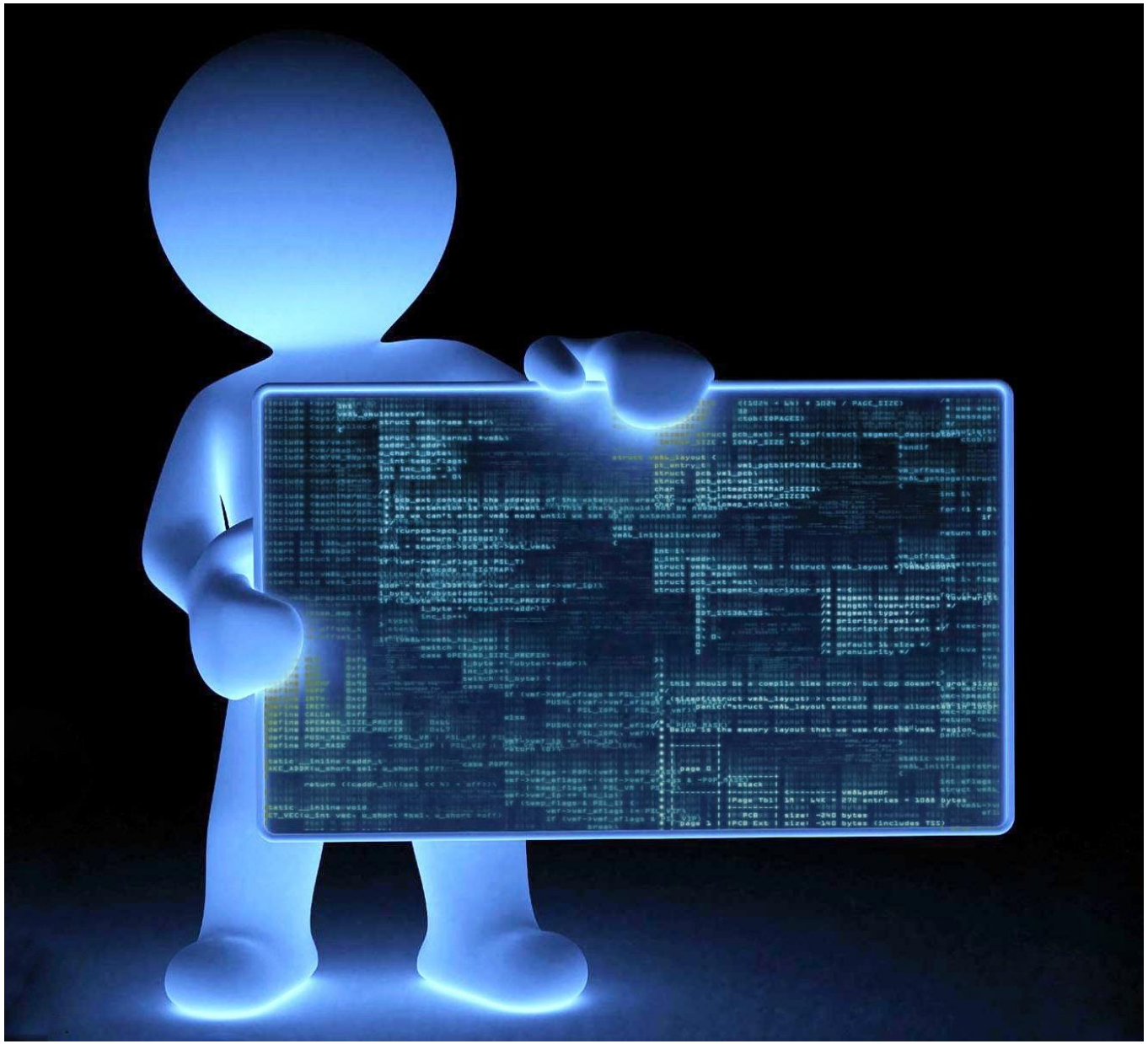
Why do we use C# for Crestron

Floating point Calculations

Access to HTTPS devices

Parsing XML/HTML/JSON

Because it’s really cool



Why do we use C# For Crestron

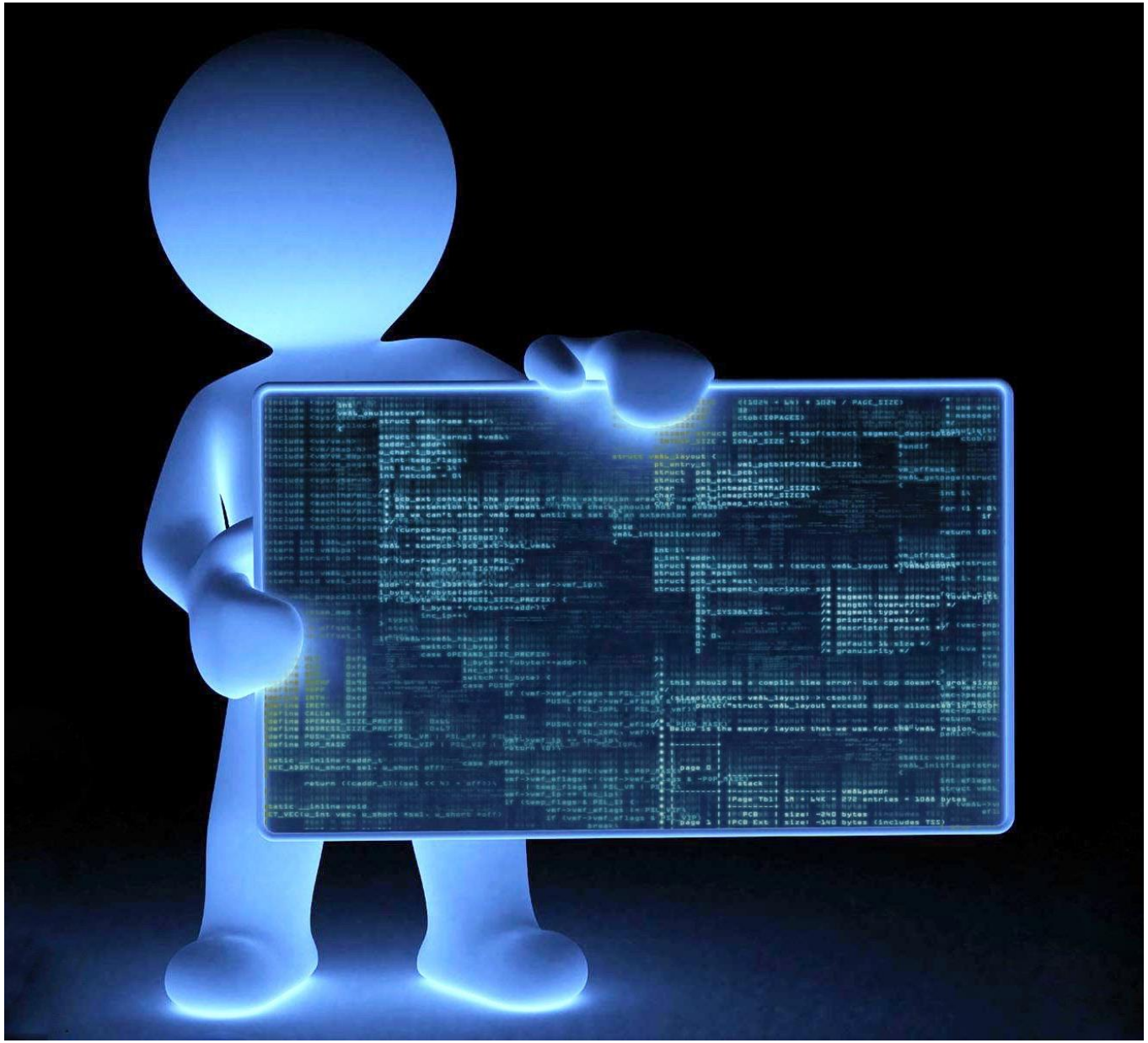
Dynamic Code

Hardware Independent

Direct Device Access

Console Commands on Processors

Threading



What's NOT in C# for Crestron

Older Crestron devices

Most existing Simpl and Simpl+ modules

However …

You can put these in a SIMPL Windows program

Interface using EISC to the other program

CRESTRON

Anatomy of a C# for Crestron Program

Public Class ControlSystem

Inherits from CrestronControlSystem

Your Program Starts here

You can only have ONE ControlSystem Class

```
1 reference
public class ControlSystem : CrestronControlSystem
{

    0 references
    public ControlSystem()
        : base()
    {
        try
        {
            Thread.MaxNumberOfUserThreads = 20;

            //Subscribe to the controller events (Sy
            CrestronEnvironment.SystemEventHandler +
            CrestronEnvironment.ProgramStatusEventHa
            CrestronEnvironment.EthernetEventHandler
        }
        catch (Exception e)
        {
            ErrorLog.Error("Error in the constructor
        }
    }

    /// <summary>
```

Anatomy of a C# for Crestron Program

Constructor

Initialize the max number of threads

Cannot Send/Receive data

Make sure it's in a try/catch

Has to exit in a timely fashion

What can you do here:

- Register Devices
- Register Event Handlers
- Add Console Commands

```
1 reference
public class ControlSystem : CrestronControlSystem
{

    0 references
    public ControlSystem()
        : base()
    {
        try
        {
            Thread.MaxNumberOfUserThreads = 20;

            //Subscribe to the controller events (Sy
            CrestronEnvironment.SystemEventHandler +
            CrestronEnvironment.ProgramStatusEventHa
            CrestronEnvironment.EthernetEventHandler
        }
        catch (Exception e)
        {
            ErrorLog.Error("Error in the constructor
        }
    }

    /// <summary>
```

Anatomy of a C# for CrestronProgram

- **InitializeSystem()**
  - Think of this as the first solution in logic
  - Make sure it's in a try/catch
  - Has to exit in a timely fashion
  - Use it to:
    - Start threads
    - Configure Com and Versiports
    - Start / Initialize socket connections
    - Send Initial device configurations

```
public override void InitializeSystem()
{
    try
    {
        this.RelayPorts[1].Register(); // claim
        this.RelayPorts[1].Close(); // Close the
    }
    catch (Exception e)
    {
        ErrorLog.Error("Error in InitializeSyste
    }
}

/// <summary>
/// Event Handler for Ethernet events: Link Up a
/// Use these events to close / re-open sockets
/// </summary>
/// <param name="ethernetEventArgs">This paramet
/// such as whether it's a Link Up or Link Down
///      Ethernet adapter this event belongs to
/// </param>
```

System Event Handlers

CrestronEnvironment.SystemEventHandler

- DiskInserted, DiskRemoved, Rebooting

CrestronEnvironment.ProgramStatusEventHandler

- Stopping, Paused, Resumed

CrestronEnvironment.EthernetEventHandler

- LinkUp, LinkDown

Event Handlers

Incoming events from devices

Incoming joins from touch screens / remotes / EISC

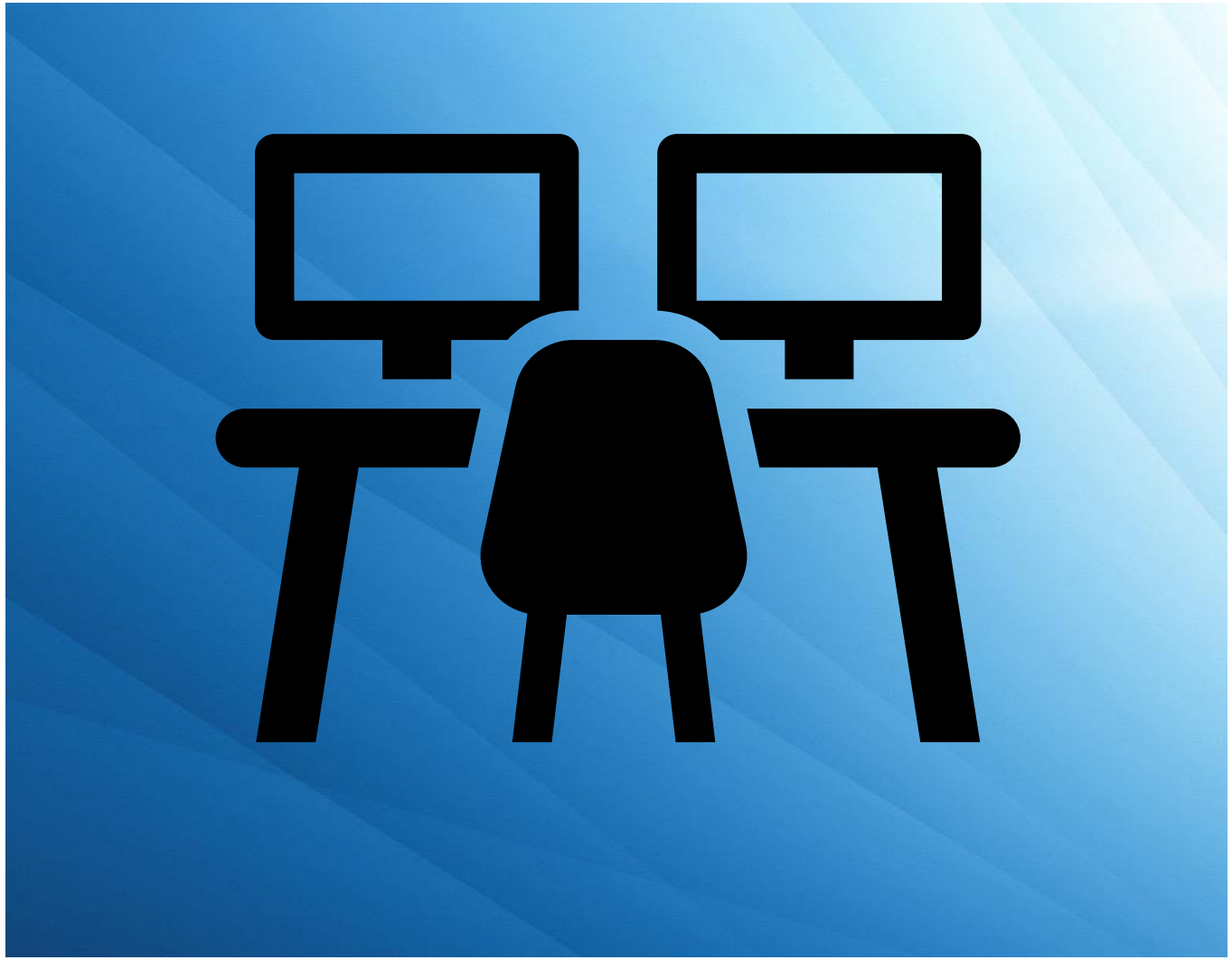
Always exit out of an event handler quickly

- **C# code is blocking. Use a thread if you need to spend any time processing**

We Can Now Start Coding

Lab Session 1

Session 1 Lab

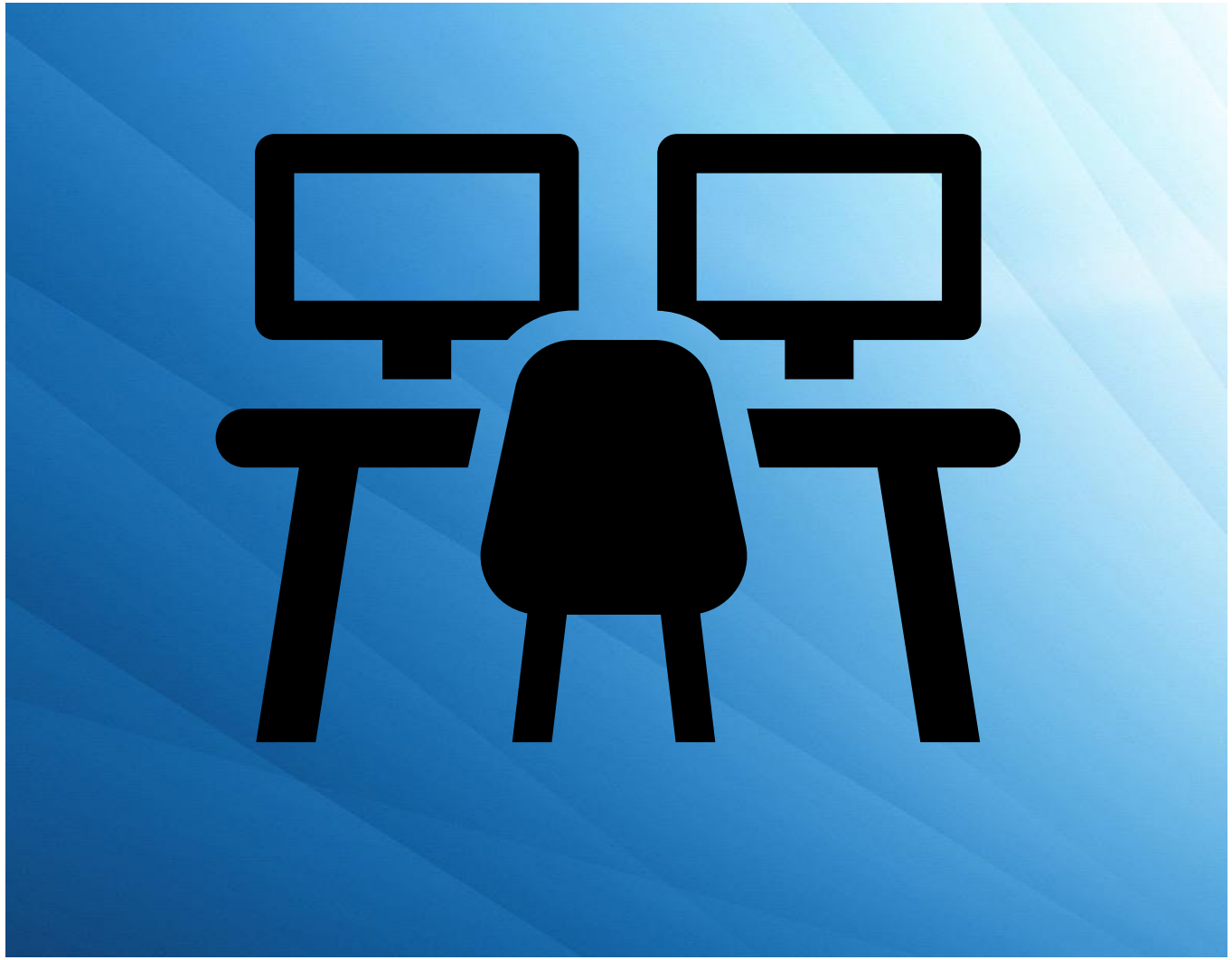


Program the Toggle, Momentary and Interlock buttons.

Complete Page Navigation Programming

Lab Session 2

Session 2 Lab

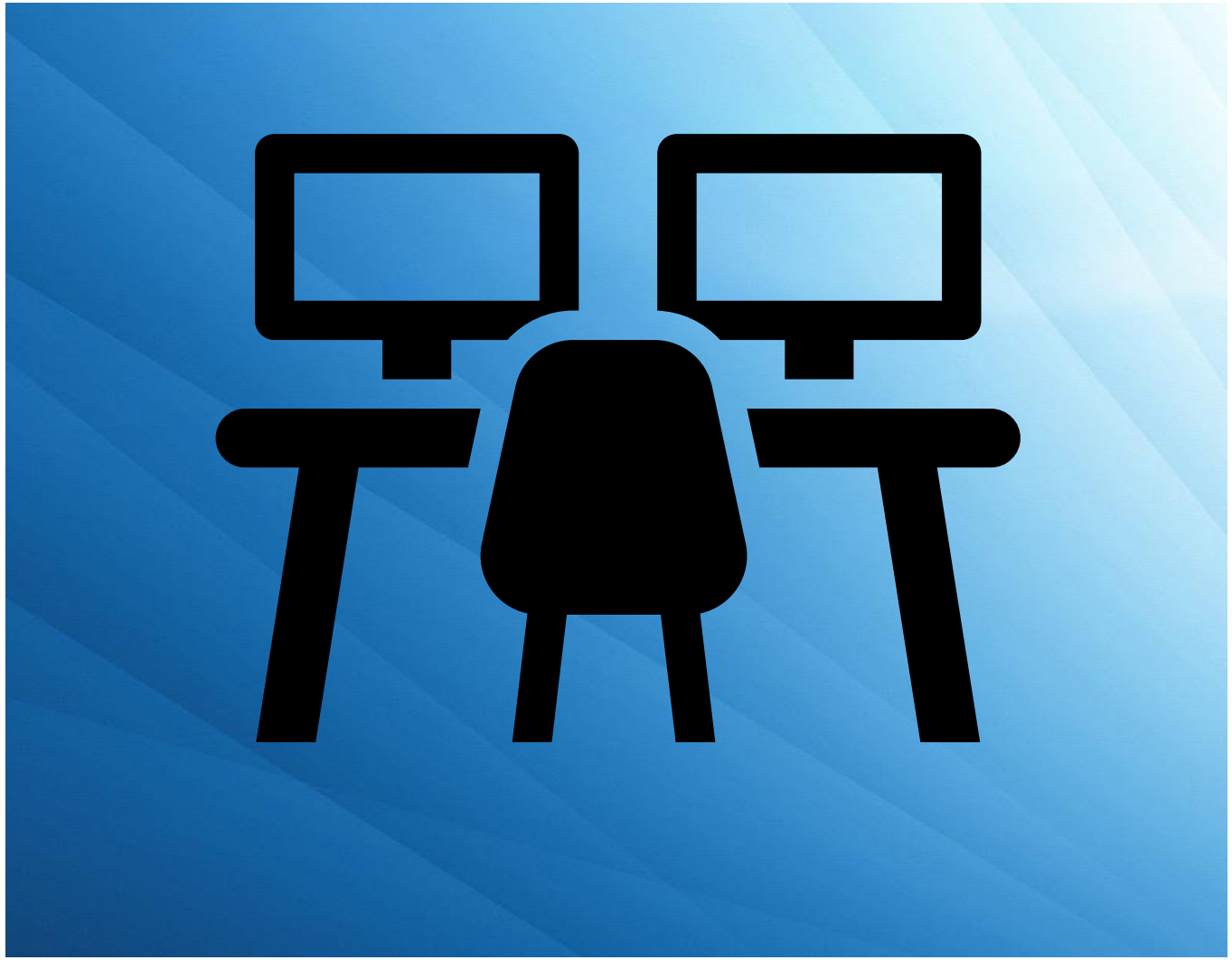


Finish Projector ON and OFF Commands.

Also handle any unexpected responses

Lab Session 3

Session 3 Lab



Sort the contents of the file we read from NVRAM and display it

Conclusion

Finished Code will be available at

https://github.com/CTI-Tim/Masters2021-MCP-101

After Masters

Thank you!