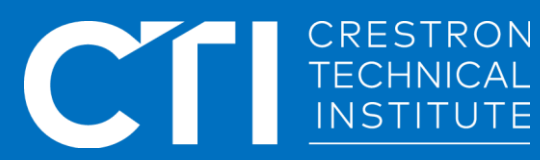




# CODING CHALLENGE

## THE GAME OF LIFE



# INTRODUCTION

Tim Gray – Sr Technical Trainer

- 30 years in the AV and automation Industry
- Programming for 40 years
- Teaching Programmers for 8 years
- Masters Instructor for 5 years

MASTERS  
CERTIFIED  
PROGRAMMER

GOLD

CTI

CRESTRON  
TECHNICAL INSTITUTE



# CONWAYS GAME OF LIFE



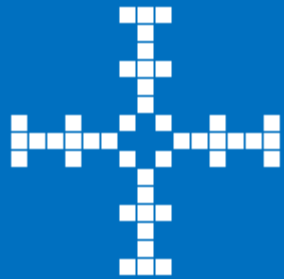
WHAT YOU NEED

<https://github.com/CTI-Tim/GameOfLifeChallenge2025>

VC4 server Instance

Write it in Simpl, S+, C#

# CONWAYS GAME OF LIFE



## HOW MUCH TIME?

You have 1 hour to attempt the challenge.

Prizes!

First and Second place



# CONWAYS GAME OF LIFE



## IMPORTANT INFO

VC4 Server Network:

Wifi : **Crestron CTI**

Pass: **Crestron2025**

Network IP Range is: **10.168.18.XXX**

No Internet on the VC4 Network

# CONWAYS GAME OF LIFE

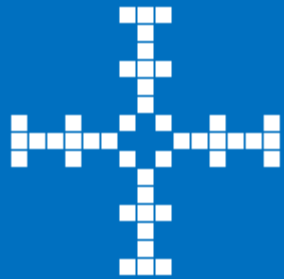


## OVERVIEW

Invented by John Horton Conway  
In 1970

This is a zero player game meaning  
that it's evolution is determined by  
it's initial state.

# CONWAYS GAME OF LIFE



## RULES OF THE GAME

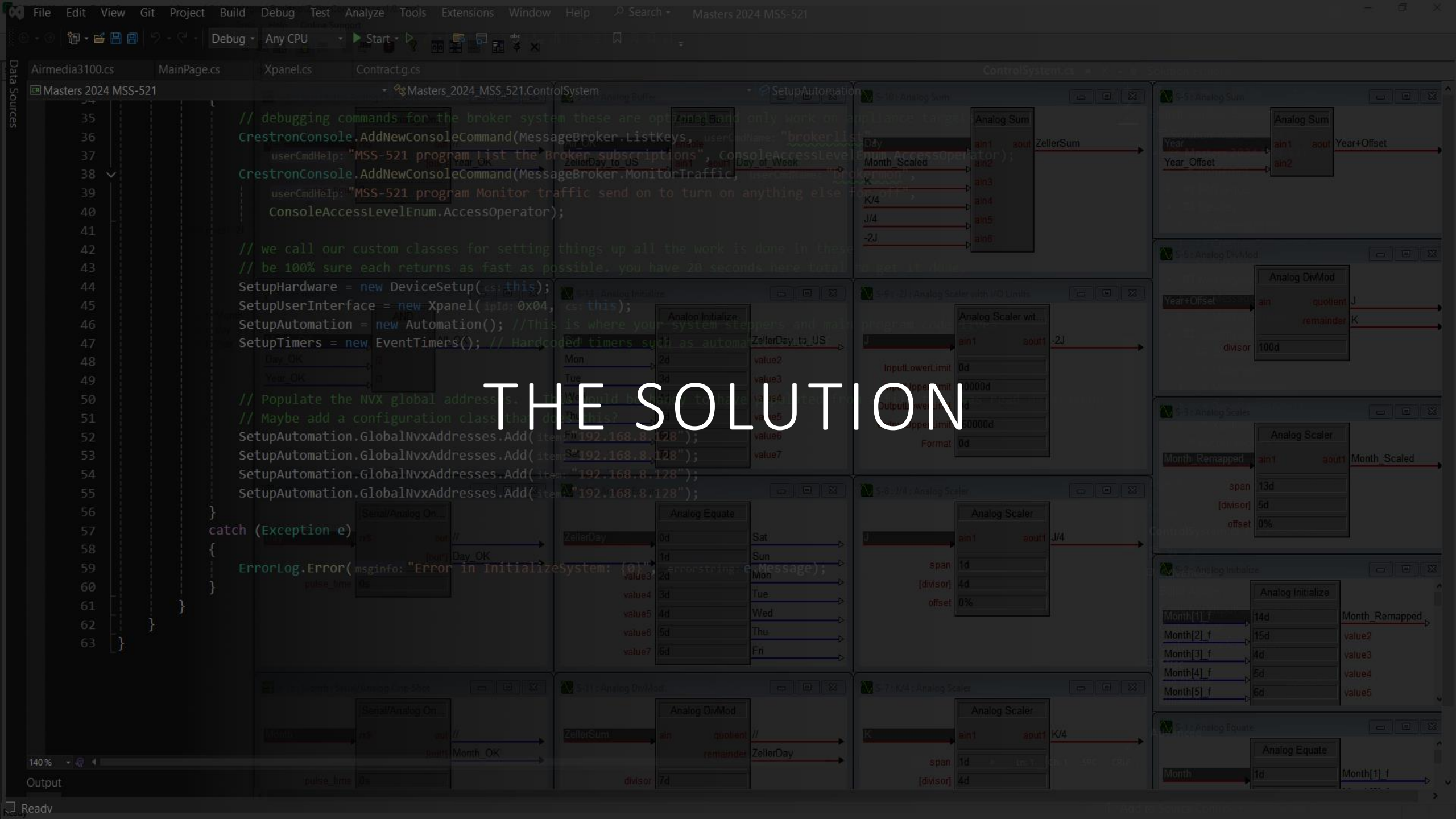
1. Living cell with less than 2 neighbors dies
2. Living cell with exactly 2 or 3 neighbors stays alive
3. Living cell with more than 3 neighbors dies
4. Dead/Empty cell with exactly 3 neighbors becomes a live cell

# CONWAYS GAME OF LIFE



QUESTIONS?

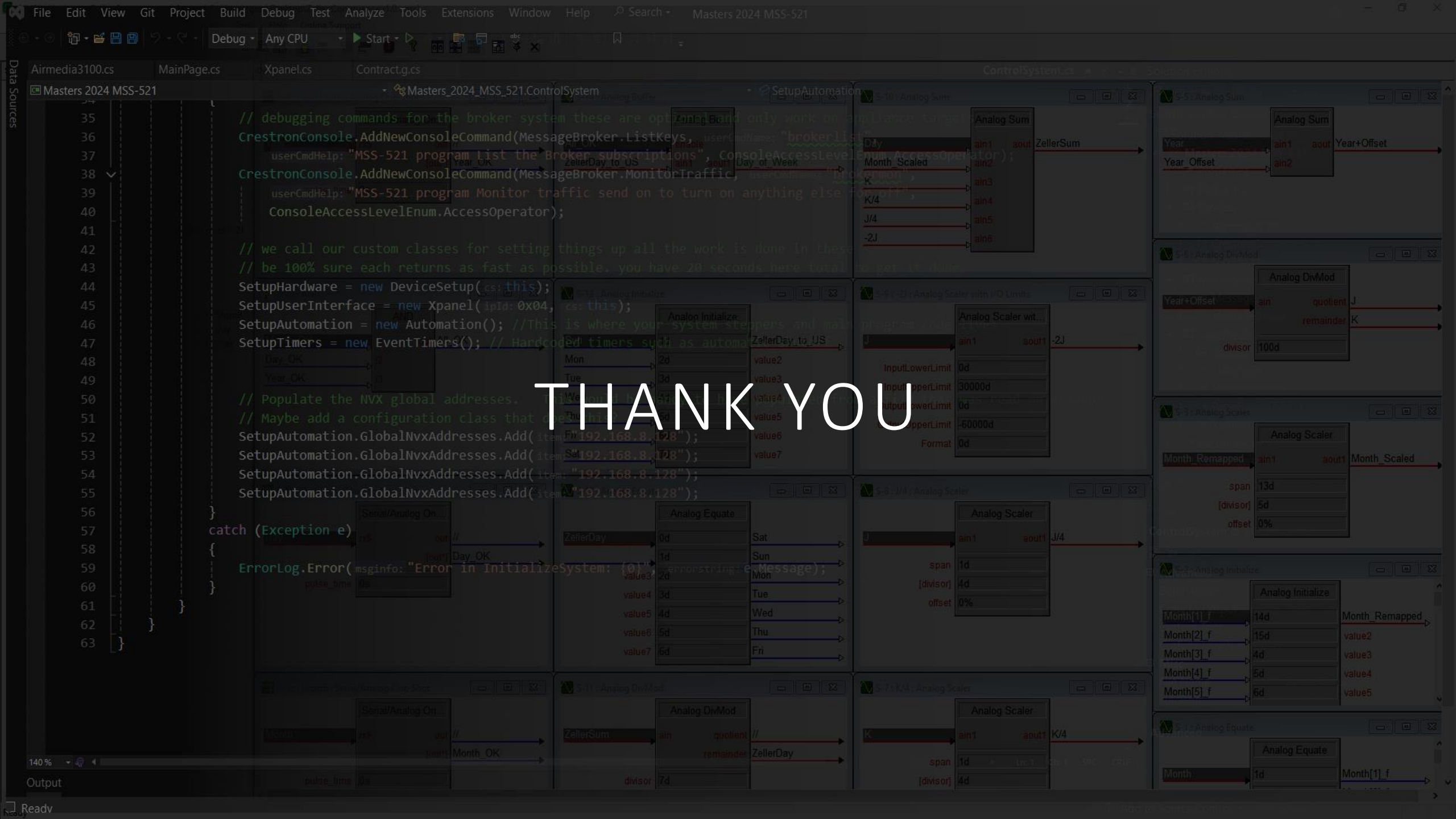




# THE SOLUTION

# IMPORTANT FACTS

1. Data grids are not easy in Simpl
2. The whole array has to be processed first
3. Updated states then pushed to the array



```
// debugging commands for the broker system these are optional and only work on an implanted device
CrestonConsole.AddNewConsoleCommand(MessageBroker.ListKeys, userCmdName: "brokerlist", consoleAccessLevelEnum.AccessOperator);
userCmdHelp: "MSS-521 program list the Broker subscriptions", consoleAccessLevelEnum.AccessOperator);
CrestonConsole.AddNewConsoleCommand(MessageBroker.MonitorTraffic, userCmdName: "brokermon", consoleAccessLevelEnum.AccessOperator);
userCmdHelp: "MSS-521 program Monitor traffic send on to turn on anything else +0x0fff", consoleAccessLevelEnum.AccessOperator);

// we call our custom classes for setting things up all the work is done in these
// be 100% sure each returns as fast as possible. you have 20 seconds here total to get it done
SetupHardware = new DeviceSetup(cs: this);
SetupUserInterface = new Xpanel(ipId: 0x04, cs: this);
SetupAutomation = new Automation(); //This is where your system steppers and main logic is
SetupTimers = new EventTimers(); // Hardcoded timers such as automating ZellerDay to US

// Populate the NVX global addresses.
// Maybe add a configuration class that does this
SetupAutomation.GlobalNvxAddresses.Add(item: "192.168.8.128");
SetupAutomation.GlobalNvxAddresses.Add(item: "192.168.8.128");
SetupAutomation.GlobalNvxAddresses.Add(item: "192.168.8.128");
SetupAutomation.GlobalNvxAddresses.Add(item: "192.168.8.128");

}
catch (Exception e)
{
    ErrorLog.Error(msginfo: "Error in InitializeSystem: (0)", errorstring: e.Message);
}
}
```

THANK YOU