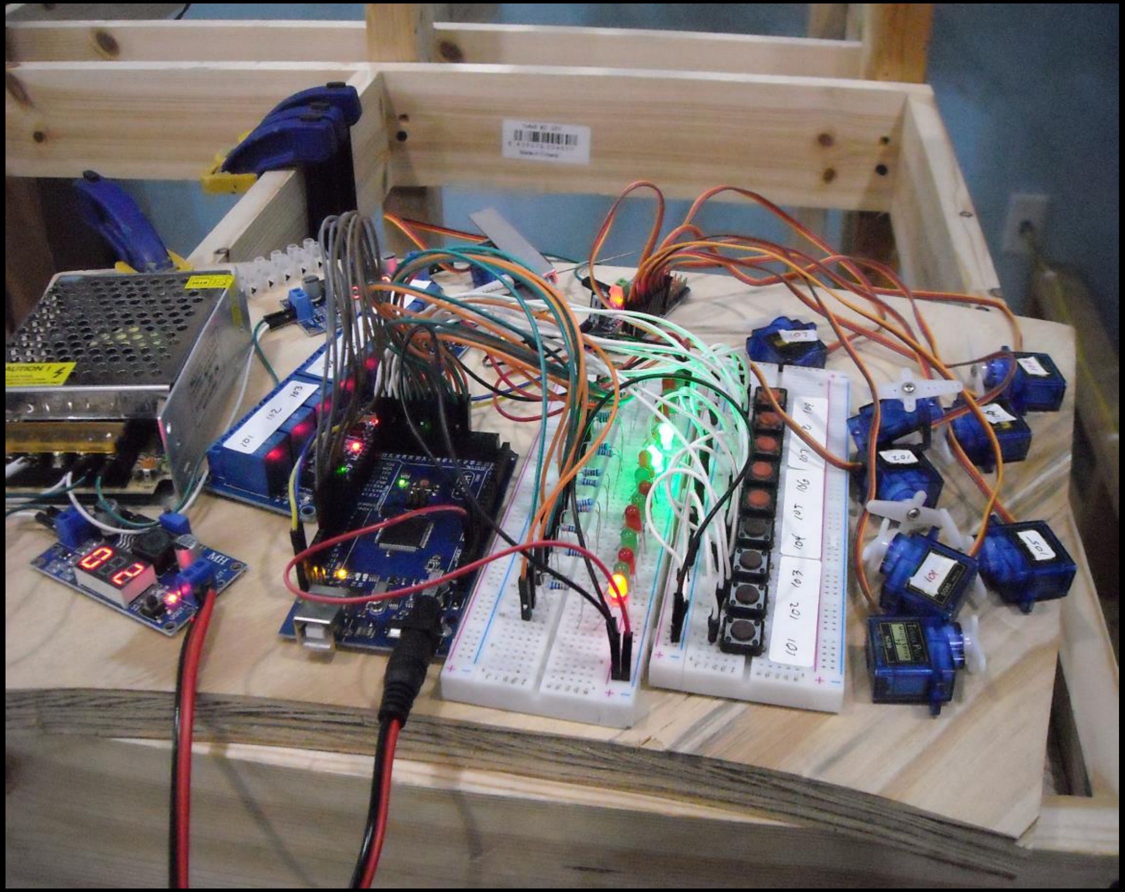
Route Control Using Arduinos

Presented By
Sandy Warrington

The Test Stand

- Route Control Is Easier Than Pushing Several Buttons
- Needed A New Test Stand
- Original Test Stand Was Disorganized
- Difficult To Move
- Wanted To See Arduino And Control Panels At Once

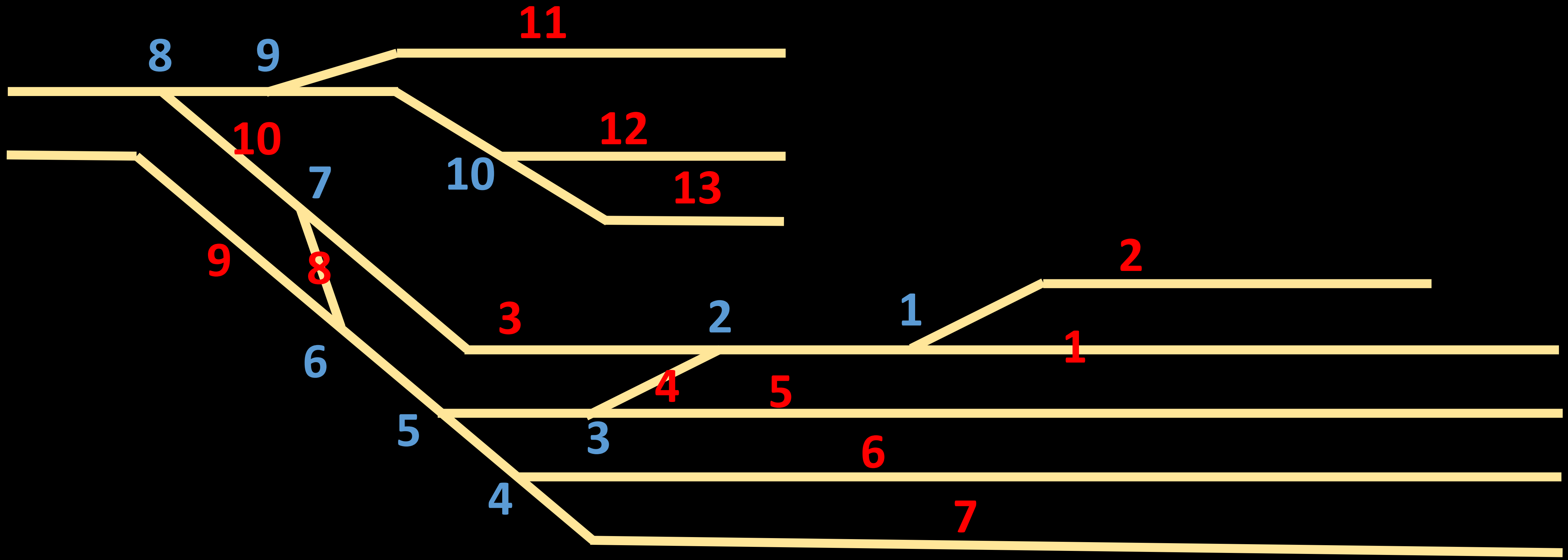


Planning

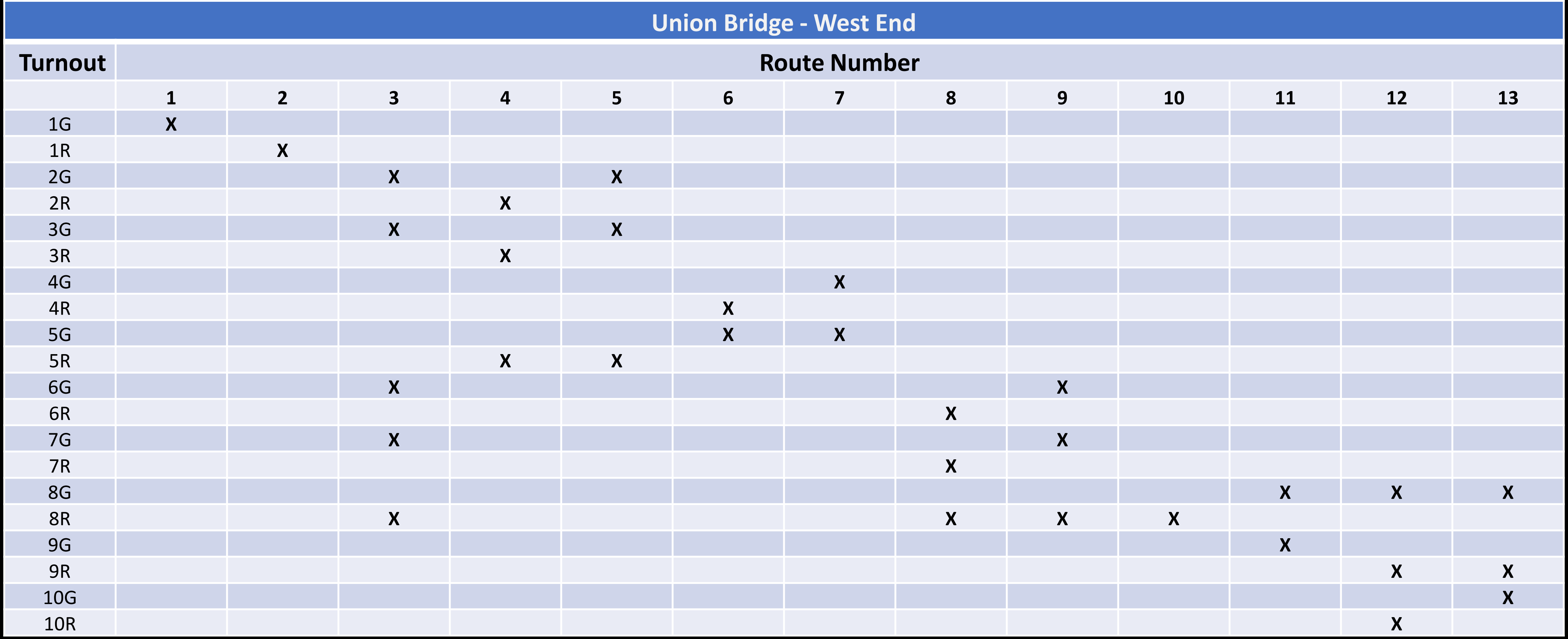
- Create A Pushbutton Turnout Control With Panel Indication On A Track Mimic
- Cost Per Turnout:
  - Using Servos: $11.52
  - Using Tortoise And Relays: $31.97
  - Using Tortoise Only: $25.19

Planning - Number Each Route And Turnout

- Turnout Numbers Are Servos And LEDs
- Route Numbers Are Pushbuttons



Planning - Route To Turnout Matrix



Union Bridge - West End													
Turnout	**Route Number**												
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**
1G	X												
1R		X											
2G			X		X								
2R				X									
3G			X		X								
3R				X									
4G							X						
4R						X							
5G						X	X						
5R				X	X								
6G			X						X				
6R								X					
7G			X						X				
7R								X					
8G											X	X	X
8R			X					X	X	X			
9G											X		
9R												X	X
10G													X
10R												X	

Hardware

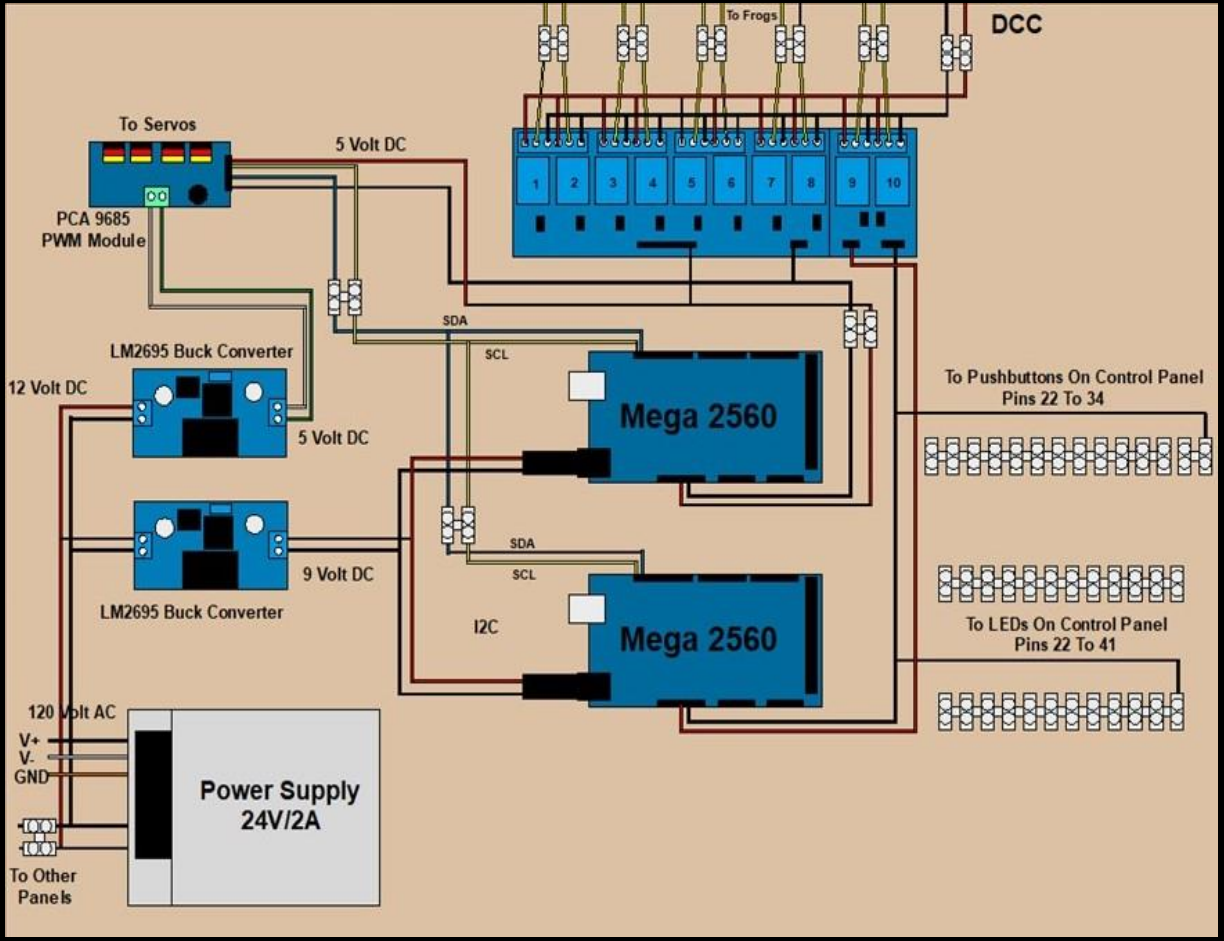
- Arduino(s) – MEGA 2560
- Power Supply And Fuse
- Buck Converter(s) To Regulate Voltage
- PCA 9685 PWM Controller For Servos
- Relays For Each Frog Requiring A Relay
- LEDs, LED Mounts, and Resisters

Hardware

- Normally Open Pushbuttons
- Dupont Connectors:
  - (Buy’em - Make’em - Substitute’em)
- Servo Extension Wires (Buy’em Or Make’um)
- Terminal Strips
- Arduino Panel To Control Panel Cable
- Plastic Tube For Standoffs And #4x5/8” Screws

Panel Construction

- Create A Board Mock-up
- Color Code Wires
- Label Wires And Terminations
- Keep It Neat! – You Can’t Troubleshoot Spaghetti
- Repeat For Control Panel



The Code

- Fear Not The Code
- Code Does Not Explode – It Just Does Crazy, Unexpected Things – Or Nothing At All
- Arduino Uses A C++ Reduced Instruction Set
- Arduino Language Reference:

www.arduino.cc/documents/reference

The Code

- A Learning Curve For The Commands And When/How To Use Them
- Keep Practicing And Start Small
- Give Pins, Functions, and Variables English Names
- Use Comments And ‘White Space’ For Clarity

The Code – General Layout

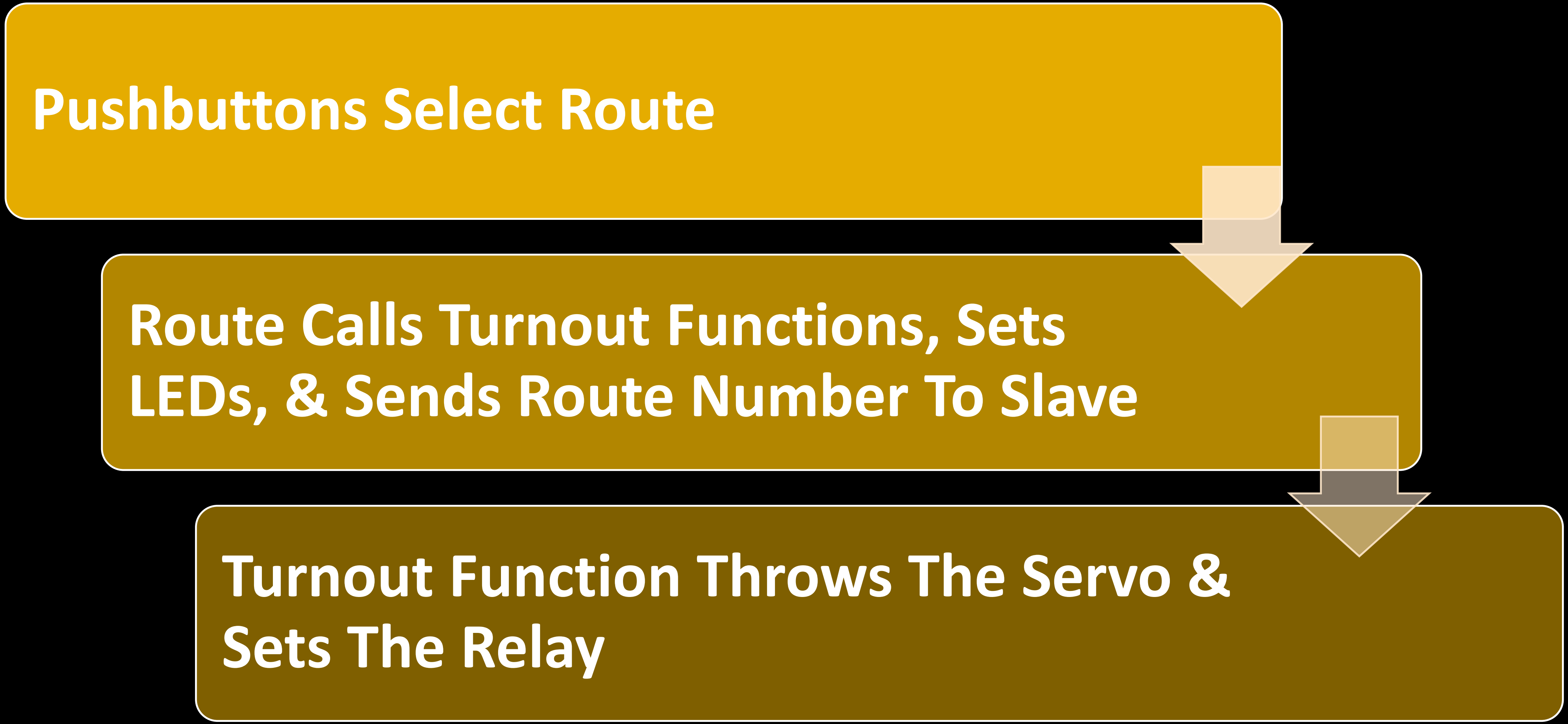
Declaration Section – Runs Once

setup() Function – Runs Once

loop() – Continuous Loop

Function Section – On Demand

The Code – Process Flow



Pushbuttons Select Route

Route Calls Turnout Functions, Sets LEDs, & Sends Route Number To Slave

Turnout Function Throws The Servo & Sets The Relay

The Code – I2C Network

- I2C Serial Communications
- Include the Wire.h Header File
- In setup() Function:
  - Start The I2C Network - Wire.begin()
    And Define The Arduino As Master Or Slave
  - Set The Baud Rate For The Network

The Code – PWM Servo Driver

- Include Adafruit_PWMServoDriver.h Header File in Declarations
- Define Global Variables For Servo Control
- Define Servo Tables
- Tables Are Zero Based – My Data Starts At 1
- In setup(): Define Servo H/L Limits And lastPos For Each Servo

Slave Arduino

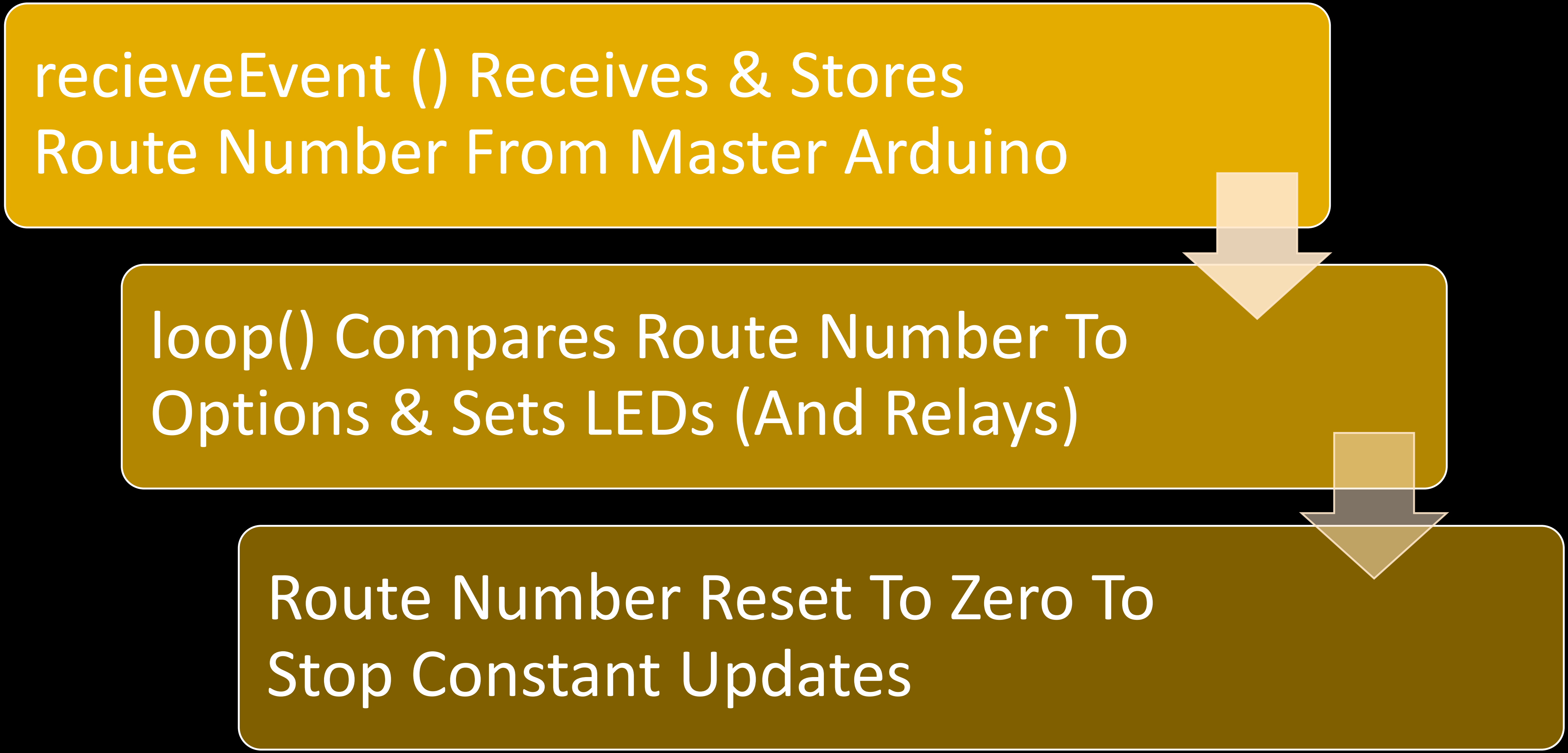
- Controls LEDs
- Controls Relays – If Necessary
- Prevents Master Arduino Overload
- Uses Wire.h Header File
- Define Variable routeNum To Store Route Number From Master

Slave - receiveEvent() Function

- Required To Receive Serial Communication From Other Devices
- Must Include A Data Type And Name For The Data Passed In The Parentheses:

```
receiveEvent(int howMany)
```

Slave – Process Flow



recieveEvent () Receives & Stores Route Number From Master Arduino

loop() Compares Route Number To Options & Sets LEDs (And Relays)

Route Number Reset To Zero To Stop Constant Updates

Discovered Problems

- Discovered Through Experience
- Both Problems Are Annoying
- Solved Through Research And Testing

Problem 1 – Servo “Bounce”

- Symptom: Servo ‘Bounces’ (Jumps) To The Opposite Limit Then Moves To The Selected Limit
- Occurs: When Repositioning The Servo But The Servo Is Already In The Desired Position
- Cause: Servo Design
  - Occurs With Both Adafruit_PWMServoDriver.h And Servo.h Header Files

Problem 1 – Servo “Bounce”

- Corrective Action:
  - Use The turnoutThrow() Function:
    - Avoids Repositioning If In The Last Position
    - Stores The Last Position Of The Servo
  - Initialize Servos To 90 Degrees In setup() Function

Problem 2 – Those Blinkin’ Relays

- Symptom: Relays Continuously Cycle HIGH And LOW Causing Relay LEDs To Blink On/Off
- Cause: Overloading The Arduino
- Corrective Action:
  - Attach Fewer Relays to An Arduino
  - Distribute The Relay Load
  - Self-Imposed Limit Of 8 Relays With Maximum Of 10 Relays

Handouts

- Thumb Drive Containing:
  - Arduino Turnout Route Control_3.pptx
  - Demo Code:
    - DemoPanel_Master.io
    - DemoPanel_Slave.io
  - Original Code:
    - AR305_UnionBridgeWest_Master.io
    - AR305_UnionBridgeWest_Slave.io
  - AR303_NewWindsor.io (servo.h Example)

Conclusion

Questions?

wewarrington@msn.com