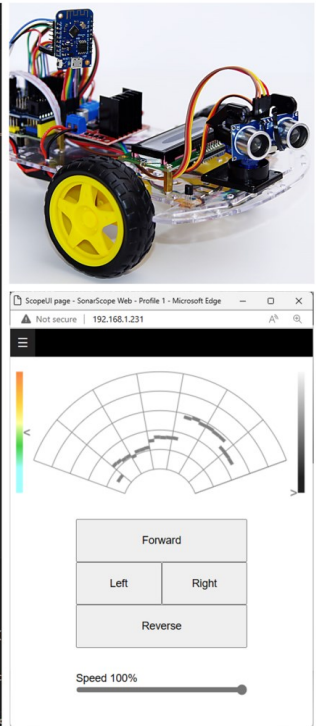


```
10 <!DOCTYPE html>
11 <html lang="en">
12 <head>
13   <meta charset="utf-8" />
14   <meta name="viewport" content="width=device-width, initial-scale=1">
15   <title>
16     <%= model->renderTitle(); %> - SonarScope Web
17   </title>
18   <link href="/Content/normal.css" rel="stylesheet">
19   <link href="/Content/ham-menu.css" rel="stylesheet">
20   <style type="text/css">
21
22     .controls {
23       margin-top: 1em;
24       margin-bottom: 1em;
25       display: flex;
26       align-items: center;
27       justify-content: center;
28     }
29
30     .item { width: 16em; }
31
32     .btn { width: 100%; height: 4em; }
33     .btn { width: 50%; height: 4em; }
34     .slider { width: 100%; }
35
36     .disable-select {
37       user-select: none;
38       -webkit-user-select: none;
39       -khtml-user-select: none;
40       -ms-user-select: none;
41     }
42   </style>
43 </head>
44 <body>
45   <nav id="hamnav">
46     <%= model->renderNav(); %>
47   </nav>
48
49   <div class="controls">
50     <div class="item">
51       <button class="btn">Forward</button>
52     </div>
53     <div class="item">
54       <div class="btn">Left</div>
55       <div class="btn">Right</div>
56     </div>
57     <div class="item">
58       <button class="btn">Reverse</button>
59     </div>
60   </div>
61
62   <div class="item">
63     <div class="slider">
64       <div class="range">
65         <div class="range__value">100%</div>
66       </div>
67     </div>
68   </div>
69 </body>
70 </html>
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL SERIAL MONITOR

SEGMENT BYTES DESCRIPTION
ICACHE 32768 reserved space for flash instruction cache
IRAM 26979 code in IROM
Code in flash (default, ICACHE_FLASH_ATTR), used 266756 / 1048576 bytes (25.4%)
SEGMENT BYTES DESCRIPTION
IROM 266756 code in flash
IntelliSense configuration updated. To manually rebuild your IntelliSense cache, click the 'Rebuild IntelliSense Project' button in the bottom right corner of the editor.
[Done] Verifying sketch 'SonarScope_Web.ino'



Arduino® Web Development

Pushing the Limits

Kashif Baig

Arduino® Web Development: Pushing the Limits

Notice of Copyright

COPYRIGHT © 2023 BY KASHIF BAIG. All rights reserved. Except as permitted by copyright laws, no part of this publication may be reproduced or distributed in any form or by any means without the prior written permission of the author, with the exception that the program listings may be entered, stored and executed in a computer system.

All trademarks or copyrights mentioned herein are the possession of their respective owners and the author makes no claim of ownership by the mention of products that contain these marks.

Disclaimer

No responsibility is assumed by the author for any injury and/or damage to persons or property as a matter of product liability, negligence or otherwise, or from any use or operation of any methods, products, instructions or ideas contained in the material herein.

Printed by:

Kashif Baig

Layout based on a free book template by Used to Tech (<https://usedtotech.com>)

First Edition, 2023

Table of Contents

| | | |
|----|--|----|
| 1. | Introduction | 1 |
| | Why Web Enable Arduino Projects? | 1 |
| | Who is this Book for?..... | 1 |
| | Introducing the MVC Web Architecture | 2 |
| | Where to get the Source Code and Tools | 4 |
| 2. | Getting Ready..... | 7 |
| | Preparing the Computing Environment..... | 7 |
| | Software Pre-requisites | 7 |
| | Arduino Boards and Components | 8 |
| 3. | Building Web Applications | 11 |
| | Hello World! A First Look at a Web Application | 11 |
| | How the Source Code is Organised | 11 |
| | Instantiating Models and Views..... | 15 |
| | Using Arduino Server Pages (ASP) for Views | 16 |
| | Authoring View ASP Files | 18 |
| | Ensuring UI Consistency with Layout Views..... | 20 |
| | Using View Helper Classes | 23 |
| | Processing HTTP Requests | 24 |
| | The Lifecycle of an HTTP Request | 24 |
| | Selecting Which View to Serve..... | 27 |
| | Handling Form Post Data with Model Binding | 29 |
| | Form Input Validation..... | 32 |
| | Handling Raw Post Data..... | 34 |

| | |
|--|----|
| Handling File Uploads | 35 |
| Serving Static Content from Program ROM | 36 |
| Serving Static Content from an SD Card | 37 |
| Downloadable Attachments | 39 |
| Returning HTTP Status Codes..... | 41 |
| Adding AJAX Enhancements..... | 42 |
| Distinguishing AJAX Requests from Regular Requests | 43 |
| User Authentication..... | 47 |
| Integrating User Authentication with a Web Application | 47 |
| Optimising Performance..... | 51 |
| Minimising the Number of Client Requests | 51 |
| Reducing Computational and I/O Overheads | 52 |
| Yielding Long Running Processes to Other Events | 53 |
| 4. Building Web APIs | 55 |
| Building Interoperable Interfaces | 55 |
| Requesting Data in a Specific Format | 56 |
| Controlling or Setting the State of a Device | 58 |
| User Authentication | 59 |
| 5. Physically Distributed Applications | 61 |
| What is a Physically Distributed Application?..... | 61 |
| Design Considerations for a Messaging Protocol | 61 |
| Message Protocol Format..... | 63 |
| Messaging Object Model | 64 |
| Project Examples Using the Messaging Library | 66 |
| Remote LED Blink | 67 |
| Remote Sensor APIs | 68 |

| | |
|---|----|
| Client Server SD Card Browser | 68 |
| 6. Testing and Debugging..... | 73 |
| Writing Testable Code | 73 |
| Fiddler, Postman and Curl | 74 |
| 7. Practical Web Applications..... | 77 |
| Scrolling Dot Matrix Display..... | 78 |
| Web Project Overview | 81 |
| Dot Matrix Display Controller Project Overview | 82 |
| RGB Light Fantastic..... | 84 |
| Web Project Overview | 87 |
| LED Controller Project Overview | 89 |
| Car with Sonar Scope..... | 90 |
| Web Project Overview | 93 |
| Car Controller Project Overview | 95 |
| Final Words | 97 |
| Appendix 1: Arduino MVC Web Framework Class Diagram | 99 |