Remote Sensing for Bridge Scour Projects – Phase 3

Ervin Sejdic (PI), Michael Rothfuss, Nicholas Franconi

RFID Center of Excellence
University of Pittsburgh
1237 Benedum Hall
3700 O’Hara St.
Pittsburgh, PA 15261

The Pennsylvania Department of Transportation
Bureau of Planning and Research
Commonwealth Keystone Building
400 North Street, 6th Floor
Harrisburg, PA 17120-0064

Technical Advisor: Paul Koza Phone: (724) 357-2845 Email: pkoza@pa.gov

16. Abstract

During Phase 2, a production ready float-out device scour monitoring system was developed based on the conceptual prototypes from Phase 1. The scour monitoring system produces a visual indication to an inspector in the vicinity of the receiver unit that scouring has occurred at a specific location and depth.

The objective of this project, Phase 3, is the modification of the Phase 2 receiver unit prototype and a field test. The modification of the Phase 2 receiver unit includes the implementation of low power LEDs on each light indicator and a solar power and battery backup system. The low power LEDs will reduce the cost of the system while increasing the lifetime of the solar panels and the battery bank. The low power LEDs will not be viewable from outside of the receiver unit without key access to the receiver unit enclosure. The objective of the field test is the deployment of the scour monitoring system and the evaluation of the installation instructions developed during Phase 2.

The field test was performed at the bridge located in Armstrong County SR1028 Segment 240. The float-out devices were installed in four (4) upstream holes spanning the river. Fifteen (15) float-out devices were installed, four (4) in three of the four holes, with about 1.5 feet between each device. The receiver unit was installed downstream within the 15-foot right-of-way. The system was extensively tested and analyzed. At all tested locations, both near, mid-way, and far from the receiver unit, float-out units successfully communicated with the receiver unit.
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Unclassified</td>
<td>Unclassified</td>
<td>116 Pages</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Form DOT F 1700.7 (8-72) Reproduction of completed page authorized
# Table of Contents

1 Executive Summary 8  
2 Phase 3 Objectives & Requirements Overview 10  
3 Modifications to Phase 2 Receiver Unit 12  
   3.1 Receiver Unit Enclosure: Solar Panel, Battery Backup System, & Antenna 13  
   3.2 Receiver Unit Design 15  
      3.2.1 Receiver Unit Power Requirements 15  
      3.2.2 Receiver Unit Hardware: Radio Frequency Receiver & Light Indicators 16  
      3.2.3 Receiver Unit Software 18  
   3.3 Receiver Unit Sealing 19  
4 Float-out Device Manufacture and Assembly 19  
   4.1 Float-out Device Software 19  
   4.2 Float-out Device Sealing 20  
      4.2.1 PVC Pipe and Custom End-Caps 20  
      4.2.2 Preparation before Sealing 21  
      4.2.3 Sealing Instructions 22  
5 Operating the Scour System 24  
   5.1 Operating the Receiver Unit 24  
   5.2 Receiver Unit and Metal Enclosure Setup 24  
      5.2.1 Accessing/Unlocking the Metal Enclosure 25  
      5.2.2 Battery Controller and Receiver Unit Power 25  
      5.2.3 Receiver Unit Power on Start-up Sequence 27  
      5.2.4 Viewing Scour Hole Number and Depth 28  
      5.2.5 Clearing Receiver Unit Scour Event Memory 30  
6 Scour System Installation Instructions & Analysis 31  
   6.1 Receiver Unit Installation & Analysis 31  
   6.2 Float-out Device Installation & Analysis 34  
7 Laboratory and Field Validation Testing 41  
   7.1 Laboratory Testing Receiver Unit and Float-out Devices 41  
      7.1.1 Receiver Unit Software Testing 41
7.1.2 Sealed Float-out Device Laboratory Testing ............... 44
7.2 Field Installation Testing Receiver Unit and Float-out Devices .... 46
  7.2.1 Field Testing the Float-out Devices .......................... 46
  7.2.2 Field Testing the Receiver Unit .............................. 47

A Receiver Unit Validation Images 50
B Float-out Device Validation Images 56
C Appendix: Receiver Unit Bill of Materials 57
D Appendix: Float-Out Device Bill of Materials, PCB, & Schematic 61
E Appendix: Receiver Unit Embedded Code 67
F Appendix: Float-out Device Embedded Code 100
### List of Figures

1. Bridge Scour Float-out Device Monitor System .......................... 10
2. The four separate scour hole locations where float-out devices were installed. 11
3. Block Diagram of the receiver unit and the receiver unit enclosure. .... 13
4. Metal enclosure, solar panel, battery backup system, antenna, and protected receiver unit ......................................................... 14
5. The receiver unit within its plastic enclosure, mounted to the backplane of the enclosure chassis. ........................................... 15
6. Receiver unit printed circuit board showing major components marked. . 17
7. Receiver unit software flow diagram. ...................................... 18
8. Float-out device identifying information. .................................. 20
9. PVC tube at right and the top and bottom custom end-caps, at left, used for each float-out device. Note the off-center magnetic reed switch mounting hole in the end-cap. ........................................... 21
10. Numbered float-out device sealing steps. Refer to steps #1-5 in Section 4.2.2 22
11. Silicone Sealant on float-out PCB edge. Refer to step #6 in Section 4.2.2 . 22
12. Numbered float-out device sealing steps. .................................. 23
13. Antenna’s white coaxial cable protruding through the cable gland port to mate to the receiver unit’s black coaxial cable. ...................... 26
14. Process of unlocking the metal enclosure and removing the lid to gain access to the receiver unit ................................................. 27
15. Locations of the toggle switches for the battery controller system (CB2) and the solar panel (CB1). .......................................... 27
16. Light indicator LED matrix showing the start-up sequence before the receiver unit is ready for operator use. ...................... 28
17. Location of the momentary pushbutton switch on the receiver unit. .... 29
18. Light indicator LED matrix showing the “N” which indicates that no float-out devices have registered with the receiver unit. .......... 29
19. Light indicator LED matrix showing four scour events. .............. 29
20. Light indicator LED matrix showing the scour event memory reset sequence. The switch must be firmly held down during the entire sequence, until the sequence completes. The sequence is complete when all LEDs turn off. . . . 30
Receiver unit installation showing all components of the installation, including: the mounting post, the antenna and shim, the solar panel, the lightning rod, and wooden supports. ............................ 32
Illustration showing the antenna mounting angle and its influence on when a float-out device comes into the receiver unit system's field-of-view. ............................ 33
Approximate installation locations for each of the four selected locations. 35
Gardner Denver articulated drill rig track mount secured to a lowboy trailer. 36
Articulating rig with geotechnical mast, tooling, and platform that are extended over the bridge and supported by a wood block. ............................ 36
Installation jig for the float-out device. ............................ 37
Installation depths of each installed float-out sensor. ............................ 39
Method of determining amount of backfill sediment to provide a desired amount of backfill between successive float-out devices within the same scour hole. Here, 12 inches of backfill is the target. ............................ 40
Removal of the magnet superglued to the PVC end-cap. Removing the magnet deactivates the anti-trigger feature, and permits the float-out device to trigger upon tilting the device. ............................ 45
Grid showing the receiver unit registering each float-out device's wireless transmission signature. ............................ 46
The two manufactured and assembled receiver unit printed circuit boards. ............................ 50
Receiver unit printed circuit board layout in ExpressPCB. ............................ 51
Receiver unit printed circuit board schematic in ExpressPCB. ............................ 52
Receiver unit current consumption from a 12 V supply. Current consumption is about 25.15 mA (i.e., 0.02515 A), which equals about 0.3018 W. The receiver will be autonomous for about 69 days at this power consumption. ............................ 53
Mounting heights of the receiver unit enclosure, antenna, and solar panel. ............................ 54
Float-out device tested about 486 feet (162 paces) downstream. ............................ 54
Float-out device tested at the first stream bend. ............................ 55
Float-out device tested about six (6) feet in downstream forward the bridge. 55
The sixteen (16) float-out device PCBs that were sealed into the capsules shown in Figure 40. ............................ 56
The sixteen (16) sealed, painted, and marked float-out devices, showing front and reverse sides. ............................ 56
Float-out device sensor PCB layout. Use for reference with the schematic. ............................ 61
Schematic of the float-out device sensor. Use for reference with the Bill of Materials.
List of Tables

1. Battery Calculations for Off-Grid Power ........................................ 16
2. Receiver Unit Validation Test Sequences. ...................................... 41
3. Receiver Unit Field Testing Results ............................................... 49
The contents of this report reflect the views of the author(s) who is(are) responsible for the facts and the accuracy of the data presented herein. The contents do not necessarily reflect the official views or policies of the U.S. Department of Transportation, Federal Highway Administration, or the Commonwealth of Pennsylvania at the time of publication. This report does not constitute a standard, specification or regulation.

This work was sponsored by the Pennsylvania Department of Transportation and the U.S. Department of Transportation, Federal Highway Administration.
1 Executive Summary

From 1966 to 2005, a total of 1,502 bridge failures occurred in the United States with approximately 60% of those failures due to hydraulic conditions, i.e. bridge scour. Bridge scour is the erosion of streambed material as a result of flow conditions surrounding abutments and piers supporting bridges. The developed scour monitoring system utilizes radio transmitters, called float-out devices, to relay scour severity and location to a central radio receiver. Multiple float-out devices are buried at various locations around a bridge structure. During a scour event, a float-out device is released due to removal of sediment around the bridge abutment and floats to the surface transmitting a unique identifier over radio frequencies. The central radio receiver located near the bridge waits for communication from each float-out device and displays the current scour depth and location via light indicators. The float-out device monitoring system provides an initial indication of scour severity for further investigation. The system is designed to allow monitoring of scouring without the need for inspectors to physically access the foundation areas during the scour event.

In this project phase, Phase 3, several modifications were made to the Phase 2 system in order to address personnel access concerns as well as to provide system autonomy through a solar power unit. Light indicator units are now only viewable when an operator has physical access to the locked metal enclosure; no light indicators are visible to the public. The solar panel and battery back-up system were included in the system design, and can provide up to 69 days of autonomy. The re-design to include light indicators only when an operator has access to the enclosure also served to reduce system power consumption and increase the number of days for autonomy. The system supports up to sixteen (16) float-out device sensors. Additionally, step-by-step instructions are provided for assembling and preparing the float-out device sensors.

Operating the system is simple. Once an operator has removed the padlock guarding the enclosure, a single pushbutton interface allows full system control. A short momentary press activates the light indicators to display the current scour status; the lights turn off after a four (4) second timeout. A long press, about 30 seconds, cycles through a light display sequence to indicate that the on-board computer’s memory of scour locations is about to be cleared. The long press and corresponding display sequence ensures that memory cannot be cleared accidentally.

The system was installed in two parts, first the float-out device sensors and second the receiver unit. During this phase, the float-out devices were installed 2 days prior to the
receiver unit installation. Future installations should attempt to install the receiver unit before the float-out devices, in order to catch any possible scour events that deploy a float-out device sensor.

The float-out device sensor installation and field testing required about seven (7) hours for four scour hole locations with four (4) float-out device sensors in each scour hole; the heavy clay composition of the streambed was responsible for lengthy drill times. It should be noted that one of the float-out device sensors could not be installed due to sediment washout in the scour hole, resulting in less space to install the sensors. However, this fact will not impede system performance, only the scour depth resolution in that hole will be lower than the other scour holes. An important consideration during the installation was the backfill sediment mixture and amount between each float-out device in each hole. About one (1) foot of a 20/30/50 mixture of fine sand/coarse sand/streambed mud was poured between successive float-out devices, and about 1.5 feet between the shallowest float-out sensor and where the streambed meets the water surface.

The receiver unit installation and field testing required about four (4) hours. The most time-consuming component of the installation was verifying the antenna placement to maximize coverage over the stream. Antenna placement did not require fine-tuning, just a simply aiming it towards the intended area for coverage. The system was tested by activating a spare float-out device in the water at various locations along the stream. The system covers from halfway underneath the bridge to about 486 feet downstream. The downstream distance was limited by personnel access to the stream. The meandering of the stream and surrounding brush did not affect float-out device detection. In future installations, if coverage is desired upstream in the event a float-out device is trapped after release but before entering the detection area of the downstream antenna, a second upstream antenna can be used concurrently.
2 Phase 3 Objectives & Requirements Overview

A prototype float-out device bridge scour monitoring system is illustrated in Figure 1 with the focus on a single substructure member. Other substructure members are simple replications of Figure 1. The installed devices, also called float-out devices, are fabricated into a watertight Radio Frequency (RF) friendly cylindrical container capable of being deployed through a 4-inch flush joint casing.

![Figure 1: Bridge Scour Float-out Device Monitor System](image)

During Phase 2, a production ready float-out device scour monitoring system was developed based on the conceptual prototypes from Phase 1. The scour monitoring system produces a visual indication to an inspector in the vicinity of the receiver unit that scouring has occurred at a specific location and depth.

The Phase 3 project objective is to modify the Phase 2 receiver unit prototype for use in a field test, to be performed with the float-out device bridge scour monitoring system. Modifying the Phase 2 receiver unit to the Phase 3 requirements includes implementing low power light emitting diodes (LED) on each light indicator and deriving power to the Phase 3 receiver unit by a solar power and battery backup system. The low power LEDs reduces the bulk of the receiver unit and reduces the cost of the system. Lower power consumption also leads to increased usable lifetime of the solar panel and battery backup system.

For the field test, PennDOT has provided the University with a bridge located in Armstrong County SR 1028 Segment 240. The float-out devices were installed at four (4) separate scour locations around the bridge abutments, shown in Figure 2 as yellow circles. The float-out devices were installed at depth increments of about two (2) feet, beginning from the bedrock below the streambed. The receiver unit was installed on the downstream side of the
bridge with the antenna facing the downstream. The low power LEDs will be visible only when the receiver unit enclosure is unlocked so that the inside of the enclosure is accessible; a button is pressed within the enclosure to activate the LEDs.

![Image of water channel with yellow circles at locations 1, 2, 3, and 4]

Figure 2: The four separate scour hole locations where float-out devices were installed.

Phase 3 was divided into three (3) primary tasks. First, the receiver units were re-designed to be compatible with the solar panel and battery backup system as well as the low power LEDs; second, the float-out devices were manufactured, software loaded with the appropriate unique identification number, bridge number, scour location, and scour depth, then assembled and painted for easy identification; revised Installation and Operation instructions were also provided; third, the field test was performed, where the receiver unit and float-out devices were installed, and the installation procedure documented.
3 Modifications to Phase 2 Receiver Unit

The receiver unit was re-designed to include the radio frequency receiver and the light indicators; both were combined into a single unit called the receiver unit, which was installed within an enclosure, called the receiver unit enclosure. The receiver unit enclosure is comprised of a metal enclosure with accompanying battery backup system, within the enclosure, and solar panel unit, external to the enclosure. Two (2) receiver units were developed for Phase 3 (see Figure 31 in Appendix A), and only one is installed within the receiver unit enclosure. The remaining receiver unit is held by PennDOT as a spare. The receiver unit supports four (4) scour locations and four (4) scour depths; a total of sixteen (16) uniquely coded float-out devices can communicate with the receiver unit. While the radio frequency receiver remains generally the same as from the Phase 2 iteration, the light indicator units are now low power LEDs, which can only be activated once the receiver unit enclosure is opened and a button is pressed on the receiver unit. A block diagram illustrating the organization of the receiver unit and receiver unit enclosure is shown in Figure 3.

The receiver unit modifications for Phase 3 are the following:

- Must be powered by a solar power and battery backup system.
- An enclosure for receiver unit that can provide protection against water, ice, and snow and contains a key lock to prevent tampering.
- Reduced size and power consumption of the receiver unit by combining the radio frequency receiver along with the light indicators
- Light indicators are viewable only once the enclosure is unlocked and opened.
Figure 3: Block Diagram of the receiver unit and the receiver unit enclosure.

3.1 Receiver Unit Enclosure: Solar Panel, Battery Backup System, & Antenna

Figure 4 shows the enclosure (i.e., with metal cover removed), battery backup system, solar panel, protected receiver unit, and antenna. The metal cover obstructs view and operation of the enclosure’s contents while in the field.

The solar panel and battery backup system is provided by SunWize Power & Battery (www.sunwizepower.com). The solar panel and battery backup system operate in unison. Within the enclosure, a battery and a maximum power point tracking (MPPT) charge controller comprise the battery backup system, which supply power to the receiver unit ele-
Figure 4: Metal enclosure, solar panel, battery backup system, antenna, and protected receiver unit.

electronics, which also resides within the enclosure. The solar panel interfaces to the MPPT controller, which allows power derived from the solar panel to charge the battery. The battery and battery backup system are protected from weather conditions, and to provide the same protection for the developed receiver unit, the receiver unit printed circuit board (PCB) was installed within a plastic enclosure (WIG-08632, by Sparkfun Electronics). The plastic enclosure was coated in a conformal silicone (422B-340G, by MC Chemicals), and a panel-mount rubber-sealed push-button switch provides a protected interface for an operator to activate the light indicators. Any gaps in the plastic enclosure, due to wires protruding through the case, were sealed with silicone sealant (DAP® Auto/Marine 100% RTV Silicone Sealant). The receiver unit, within its plastic enclosure, is bolted to the backplane of the enclosure chassis (Figure 5). Figure 21 in Section 6.1 shows the receiver unit enclosure mounted to a pole on the bridge with for the field test. The pole mount provides elevation for the receiver unit’s mounted antenna to maximize line-of-sight between the antenna and the float-out devices. Additionally, the pole provides elevation to increase the solar panel’s exposure to sunlight.
3.2 Receiver Unit Design

3.2.1 Receiver Unit Power Requirements

Without a dedicated AC power source, power limitations must be imposed on the receiver unit, which derives its power from a battery that gets recharged by a solar panel. In order to determine the minimum required battery capacity and solar panel configuration, the current draw and days of system autonomy must be known, for phase 3 this is 60 days. The depth of discharge is a divider that provides a safety factor to prevent over-discharging the battery bank, in this case, the battery will not be discharged more than 50%. As the depth of discharge for the battery bank increases, the number of charge cycles decreases due to the chemical stresses of a complete discharge and charge cycle. Although lower ambient temperatures allow for increased battery lifetimes, the capacity of the battery bank is reduced proportionally. This effect is designed for using the battery temperature multiplier. The adjusted sun hours accounts for the misalignment between the sun and the solar panel throughout the day. Because of this misalignment, the solar panel will only output the maximum power during a short period during the day. The sun hours are a constant factor ranging from 3.5 to 6 hours, depending on the location. However, the worst case scenario assumed for phase 3 is 2 hours of sunlight per day. The solar panel capacity output per day must be greater than the receiver power (in Ah/day) consumed per day.

To provide 60 days of autonomous power (i.e., without sunlight, which may result from debris, such as leaves or snow, covering the solar panel for an extended period), 0.346 W is
available, continuously, at 12 V. In other words, about 0.02885 A can be drawn continuously from the 12 V supply. The receiver designed for phase 3 consumes about 0.02515 A continuously from a 12 V supply (i.e., 0.3018 W), and thus almost 69 days of autonomy is expected from the calculations. Table 1 summarizes the above discussion. Note the highlighted system autonomy in number of days.

<table>
<thead>
<tr>
<th>Table 1: Battery Calculations for Off-Grid Power</th>
</tr>
</thead>
<tbody>
<tr>
<td>Receiver Unit - Voltage</td>
</tr>
<tr>
<td>Receiver Unit - Current</td>
</tr>
<tr>
<td>Receiver Unit - Power</td>
</tr>
<tr>
<td>Receiver Unit - Amp-Hour per Day</td>
</tr>
<tr>
<td>Receiver Unit - System Autonomy</td>
</tr>
<tr>
<td>Battery Bank - Depth of Discharge</td>
</tr>
<tr>
<td>Battery Bank - Temperature Multiplier</td>
</tr>
<tr>
<td>Battery Bank - Nominal Voltage</td>
</tr>
<tr>
<td>Battery Bank - Capacity</td>
</tr>
<tr>
<td>Solar Panel - Wattage</td>
</tr>
<tr>
<td>Solar Panel - Voltage @ Max Power</td>
</tr>
<tr>
<td>Solar Panel - Current @ Max Power</td>
</tr>
<tr>
<td>Solar Panel - Sun Hours per Day</td>
</tr>
<tr>
<td>Solar Panel - Multiplier</td>
</tr>
<tr>
<td>Solar Panel - Adjusted Sun Hours per Day</td>
</tr>
<tr>
<td>Solar Panel - Life-time Multiplier</td>
</tr>
<tr>
<td>Solar Panel - Output Capacity</td>
</tr>
</tbody>
</table>

3.2.2 Receiver Unit Hardware: Radio Frequency Receiver & Light Indicators

The receiver unit PCB is shown in Figure 6. The major components of the unit are marked in the figure. Note the 4 x 4 array of light indicator LEDs on the left-hand side of the PCB; these are the LEDs used to indicate scour level and depth. The receiver unit PCB is manufactured using the 4-layer MiniBoard Pro process by ExpressPCB; the board dimensions are 3.8 inch x 2.5 inch. At each of the four corners, 0.251 inch mounting holes are placed, which allow the PCB to be bolted onto the receiver unit enclosure metal backplane. The schematic for the PCB and the PCB layout are shown in Appendix A in Figures 33 and 32, respectively.
Each column of the 4 x 4 LED matrix represents a scour hole. Starting with the topmost LED in a given scour hole, the top LED is green, the second LED from the top is yellow, the third LED from the top is orange, and the bottom LED is red (see Figure 3). The colors indicate the scour severity; green is the least severe, yellow the next most severe, orange is second-to-last more severe, and red is the most severe scour depth. The antenna cable connector is an SMA female jack, which mates with the pole-mounted antenna’s SMA male connector. The push-button on the PCB allows an operator to illuminate the LED matrix for viewing scour severity and location, and it allows an operator to clear the receiver unit’s memory, if desired.

The radio frequency receiver comprises of a microcontroller (MCU) unit that incorporates a radio and a microcontroller/central processing unit. The MCU executes software to handle power-up, peripheral initialization, operator interfacing and commands, radio communication, false-out device discrimination, and targeting specific light indicator LEDs. The MCU’s radio operates in the 915 MHz industrial, scientific & medical (ISM) band.

The MCU interfaces to two peripherals: (1) a voltage level-shifter for interfacing the MCU to a 16-bit input/output (I/O) expander and (2) a push-button switch for operator control. The level-shifter converts the MCU’s 3V serial peripheral interface (SPI) level signals to 5V level signals. The 5V SPI signals activate any number of the individually addressable I/O channels to target specific light indicator LEDs for illumination. The I/O expander channels feed to LED driver circuitry to provide sufficient illumination for viewing in both bright and dark ambient light conditions. The push-button switch registers a button press...
from an operator. Both momentary and long button presses are recognized and control the receiver unit accordingly. Please see Section 3.2.3 for details on receiver unit control via the push-button.

For a list of the specific components used in the receiver unit design, please see Appendix C for the bill of materials.

### 3.2.3 Receiver Unit Software

The majority of the receiver unit software is inaccessible to an operator. However, an operator can put the receiver unit into several software states that allow the operator to illuminate the light indicator LEDs and clear the receiver unit’s memory of scour location and severity. Figure 7 illustrates the major software states and observable effects upon executing various software states.

![Diagram](image)

**Figure 7:** Receiver unit software flow diagram.

For instructions on operating the receiver unit, see Section 5.1, and for the for the source code loaded onto the receiver unit’s MCU, please see Appendix E.
3.3 Receiver Unit Sealing

While the battery controller within the metal enclosure is resistant to debris and moisture, the bare receiver unit circuit board is not. To make the receiver unit resistant to debris and moisture, the receiver unit PCB is enclosed within a project box, and all interfaces through the project box are sealed with silicone sealant (*DAP® Auto/Marine 100% RTV Silicone Sealant*).

4 Float-out Device Manufacture and Assembly

The float-out devices were designed and developed in Phase 2, therefore details regarding their design will be omitted. A Bill of Materials, PCB layout, and schematic of a float-out device can be found in Appendix D. This section serves to detail the manufacture and assembly process to make the float-out devices ready for installation.

4.1 Float-out Device Software

Each float-out device is programmed with a scour hole number, a scour color depth indicator, the bridge identification number, and a unique identification number. Figure 8 shows a 4 x 4 matrix of cells, with each cell containing, from the top of each cell, the device number (e.g., #10), the scour depth color indicator (e.g., “YELLOW”), the scour location number (e.g., “LOC 3”), and the device’s unique identification number (e.g., 770). Each cell in the 4 x 4 matrix of Figure 8 corresponds to a respective LED location in the receiver unit’s 4 x 4 LED matrix (see Figures 3 and Figure 6). Neither the bridge identification number nor the unique identification number is visible to an operator; this information is strictly used by the receiver unit’s software to reject or accept wireless signals that it receives, so that only registered float-out devices may be visible to the operator via the receiver unit LED matrix. Additionally, the bridge identification number for the bridge located in Armstrong County SR 1028 Segment 240 is **03102802400395**.

The software on each float-out device is the same as from Phase 2, so a detailed explanation of the software states will be omitted. Rather, see Appendix F for the source code loaded onto the float-out devices, as well as for an explanation for how to uniquely program each float-out device.
### 4.2 Float-out Device Sealing

#### 4.2.1 PVC Pipe and Custom End-Caps

The custom end-caps in Phase 3 are the same as in Phase 2, with the exception that the Phase 3 caps have an off-center magnetic reed switch mounting hole. The off-center hole provides room for the battery inset to the cap. The PVC pipe’s outer dimensions measure 2.9 inch diameter by 7.28 inch in length. The inner diameter is not specified by the manufacturer; however, 2.44 inch is a good estimate of the expected inner diameter. As such, some end-caps may require their outer diameter to be sanded down slightly to fit inside the PVC tube. Figure 9 shows a float-out device’s PVC tube along with its two end-caps. Note the

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>#5</td>
<td>#9</td>
<td>#13</td>
<td></td>
</tr>
<tr>
<td>GREEN</td>
<td>GREEN</td>
<td>GREEN</td>
<td>GREEN</td>
<td></td>
</tr>
<tr>
<td>LOC 1</td>
<td>LOC 2</td>
<td>LOC 3</td>
<td>LOC 4</td>
<td></td>
</tr>
<tr>
<td>257</td>
<td>513</td>
<td>769</td>
<td>1025</td>
<td></td>
</tr>
<tr>
<td>#2</td>
<td>#6</td>
<td>#10</td>
<td>#14</td>
<td></td>
</tr>
<tr>
<td>YELLOW</td>
<td>YELLOW</td>
<td>YELLOW</td>
<td>YELLOW</td>
<td></td>
</tr>
<tr>
<td>LOC 1</td>
<td>LOC 2</td>
<td>LOC 3</td>
<td>LOC 4</td>
<td></td>
</tr>
<tr>
<td>258</td>
<td>514</td>
<td>770</td>
<td>1026</td>
<td></td>
</tr>
<tr>
<td>#3</td>
<td>#7</td>
<td>#11</td>
<td>#15</td>
<td></td>
</tr>
<tr>
<td>ORANGE</td>
<td>ORANGE</td>
<td>ORANGE</td>
<td>ORANGE</td>
<td></td>
</tr>
<tr>
<td>LOC 1</td>
<td>LOC 2</td>
<td>LOC 3</td>
<td>LOC 4</td>
<td></td>
</tr>
<tr>
<td>259</td>
<td>515</td>
<td>771</td>
<td>1027</td>
<td></td>
</tr>
<tr>
<td>#4</td>
<td>#8</td>
<td>#12</td>
<td>#16</td>
<td></td>
</tr>
<tr>
<td>RED</td>
<td>RED</td>
<td>RED</td>
<td>RED</td>
<td></td>
</tr>
<tr>
<td>LOC 1</td>
<td>LOC 2</td>
<td>LOC 3</td>
<td>LOC 4</td>
<td></td>
</tr>
<tr>
<td>260</td>
<td>516</td>
<td>772</td>
<td>1028</td>
<td></td>
</tr>
</tbody>
</table>

Figure 8: Float-out device identifying information.
off-center hole for mounting the magnetic reed switch.

Figure 9: PVC tube at right and the top and bottom custom end-caps, at left, used for each float-out device. Note the off-center magnetic reed switch mounting hole in the end-cap.

4.2.2 Preparation before Sealing

Before the float-out device can be permanently sealed within the PVC tube, the float-out electronics and battery must first be installed within one of the PVC tube’s two end-caps. The following steps should be followed in order to prepare the end-cap mounted electronics for installation. When following these steps, refer to Figure 10.

1. The magnetic reed switch is mounted in its mounting hole (see Figure 9) and a liberal application of Amazing Goop® All Purpose is used to secure it in place.

2. Immediately after step 1, the battery, in its holder, is pressed firmly into theAmazing Goop® All Purpose.

3. Solder the battery holder’s wires to ground and power vias on the float-out PCB.

4. Tack the foam to PCB with generic superglue.

5. Deposit silicone sealant (DAP® Auto/Marine 100% RTV Silicone Sealant) on the inside lip of the end-cap.

6. Mount on end-cap and apply silicone sealant (DAP® Auto/Marine 100% RTV Silicone Sealant) to the outside rim of PCB (see Figure 11).
4.2.3 Sealing Instructions

To seal the float-out device electronics within the PVC tube, the following steps should be performed; Figure 12 illustrates the step numbers as they are performed. Note that prior
to installing the end-cap mounted float-out device, the other end-cap must be installed into one end of the PVC tube, using the same PVC cement in steps 1 and 2.

1. Quickly and liberally apply Purple Primer (Oatey Purple Primer - NSF Listed) to the inside of the PVC tube, beginning at the end of the tube and up to about two (2) inches from the end. The same should then be done with PVC cement (Oatey PVC Medium Clear Cement).

2. Quickly and liberally apply Purple Primer (Oatey Purple Primer - NSF Listed) to the outside of the end-cap of the end-cap mounted float-out device. The same should then be done with PVC cement (Oatey PVC Medium Clear Cement).

3. Partially insert the end-cap mounted float-out device from step 1 into the PVC tube from step 2. Then place the result into the clamp (JET Parallel Clamp 70424), ensuring that the magnetic reed switch’s magnet is not in contact with the clamp, and begin to tighten the clamp.

4. The result of tightening the clamp will appear as in this step.

Figure 12: Numbered float-out device sealing steps.
5 Operating the Scour System

5.1 Operating the Receiver Unit

The light indicator LEDs on-board the receiver unit are operated only when access is granted to the metal enclosure. With access to the receiver unit, an operator uses the receiver unit’s pushbutton and light indicator LED matrix to view float-out devices that deployed due to a scour event, their scour hole number and scour depth, and to clear the receiver unit’s memory of deploy float-out devices. Additionally, the main power to the battery controller and receiver unit can be turned on/off. Below are two step lists, the first showing the initial receiver unit and metal enclosure set-up, the second step list shows a typical operating procedure. Following the lists are the sections which provide details of the listed steps.

Initial Receiver Unit Set-up, assume metal enclosure is unlocked:

1. Ensure that the coaxial cable coming from the receiver unit and the coaxial cable of the antenna are mated, and that the coaxial cable gland is tightened.

2. Power on the battery controller & solar panel

3. Receiver unit will power itself on; wait until after the receiver unit has finished displaying its start-up sequence

4. The operator may now view scour hole number and depth or clear the receiver unit scour event memory

Typical Operating Procedure:

1. Unlock the metal enclosure and remove lid

2. Momentarily press the pushbutton switch to view scour hole number and depth.

3. If desired, clear the receiver unit’s scour event memory

4. Close lid onto the metal enclosure and lock the enclosure.

5.2 Receiver Unit and Metal Enclosure Setup

Figure 4, shown previously, shows the metal enclosure, the metal enclosure’s lid, and the receiver unit mounted within the enclosure. To ensure that the receiver unit can communicate
with float-out devices, the black coaxial cable protruding from the receiver unit must be mated to the white coaxial cable of the antenna. In order to mate the two, the white coaxial cable must first protrude through cable gland at the left-side of the metal enclosure. Once through the cable gland, mate the black coaxial cable to the antenna’s white coaxial cable, and then tighten the cable gland to prevent debris from coming through the cable gland port. Figure 13 shows the antenna’s cable white coaxial cable mated to the black coaxial cable, through the cable gland port; additionally, the solar panel’s electrical wires are shown at left in the figure.

5.2.1 Accessing/Unlocking the Metal Enclosure

Figure 14 shows the location of the metal enclosure’s lock. After the lock as been removed, the lid pulls off.

5.2.2 Battery Controller and Receiver Unit Power

Figure 15 shows the location of the main toggle switches to turn on/off power through the battery controller system (i.e., right-hand side) and power from the solar panel (i.e., left-hand side). The following order must be observed when powering on the system, and to turn off power to the system, follow the reverse order of steps:

1. Toggle the right-hand side switch, labeled CB2, to turn on the battery controller system and apply power to the receiver unit.

2. Toggle the left-hand side switch, labeled CB1, to turn on the solar panel.
Figure 13: Antenna’s white coaxial cable protruding through the cable gland port to mate to the receiver unit’s black coaxial cable.
Figure 14: Process of unlocking the metal enclosure and removing the lid to gain access to the receiver unit.

Figure 15: Locations of the toggle switches for the battery controller system (CB2) and the solar panel (CB1).

5.2.3 Receiver Unit Power on Start-up Sequence

When power is applied to the receiver unit, the receiver unit will indicate that it is ready for operator use by flashing an alternating pattern of LEDs on the LED matrix. The start of the flashing patterns will begin six (6) seconds after power is applied to the receiver unit, and
exactly four (4) LED patterns will flash. Figure 16 shows the alternating pattern sequence indicating the receiver unit is ready for use. **Do not press the pushbutton switch before the flashing sequence has completed.**

![Image of LED flashing sequence](image)

Figure 16: Light indicator LED matrix showing the start-up sequence before the receiver unit is ready for operator use.

### 5.2.4 Viewing Scour Hole Number and Depth

To view the scour hole number and scour color depth indicator for any deployed float-out devices, a pushbutton switch (see Figure 17) functions as the operator interface. The pushbutton should only be momentarily pressed to view the illuminated light indicator LEDs. The LEDs will stay lit for four (4) seconds after the pushbutton switch is first actuated. Figure 18 shows the result of pressing the pushbutton switch when no float-out devices have registered with the receiver unit (i.e., wirelessly transmitted a signal that was received by the receiver unit). The “N” indicates that **NO** float-out devices have been registered. Figure 19 shows the LED matrix displaying a scour event registered in hole numbers #9, #10, #11, and #12 at the “ORANGE” scour depth color. Note that only the deepest scour color indicator will illuminate for a given scour hole number. For example, in scour hole number 3, if the deepest scour level float-out device, corresponding to a red LED, is registered before a less severe scour level float-out device, corresponding to a yellow LED, then the yellow LED will not be displayed. This method of conserving LED usage is both intuitive (i.e., shallower float-out devices are assumed to have been registered if a deeper float-out device has been registered) and extends the operational autonomy of the system by conserving battery power.

Note that in the unlikely event that the LED matrix LEDs do not automatically turn off after four (4) seconds, simply momentarily press the pushbutton switch again; the LED matrix will automatically turn off afterward.
Figure 17: Location of the momentary pushbutton switch on the receiver unit.

Figure 18: Light indicator LED matrix showing the “N” which indicates that no float-out devices have registered with the receiver unit.

Figure 19: Light indicator LED matrix showing four scour events.
5.2.5 Clearing Receiver Unit Scour Event Memory

The field test and installation process requires that each float-out device be tested (i.e., shown that it can communicate with the receiver unit) before installation. It is therefore necessary to clear the receiver unit’s memory of the field test and installation process for capturing future scour events. Once the scour event memory has been cleared, pressing the pushbutton switch will simply display the “N,” indicating no float-out devices have been registered since the memory clear (i.e., see Section 5.2.4). Figure 20 shows the operation procedure for clearing scour event memory, and the following list describes the steps, shown in the figure, in more detail.

![Figure 20: Light indicator LED matrix showing the scour event memory reset sequence. The switch must be firmly held down during the entire sequence, until the sequence completes. The sequence is complete when all LEDs turn off.](image)

1. Hold the pushbutton down firmly. If the switch is not held down firmly, the switch will break contact, and the microcontroller unit logic will interpret the holding of the pushbutton as a momentary press to display the scour level and depth. If this happens, simply release the pushbutton and again, firmly, hold the pushbutton down.

2. Continue to hold the pushbutton down, and after ten (10) seconds, a sequence of LEDs on the light indicator LED matrix will begin to display.

3. Continue to hold the pushbutton down; the entire light indicator LED matrix will eventually illuminate and then turn off. Each LED takes one second to turn on, for a total of sixteen (16) seconds to fill the entire LED matrix.

4. Once the light indicator LED matrix turns off (i.e., after 26 seconds, from step 2 and 3), release the pushbutton. The scour event memory has now been cleared. If the pushbutton is momentarily pressed after the memory has been cleared, an “N” will display on the light indicator LED matrix indicating that no float-out devices have registered with the receiver unit.
6 Scour System Installation Instructions & Analysis

The float-out devices were installed on November 3, 2015, and the receiver unit (i.e., including the metal enclosure, battery backup system, antenna, and solar panel) was installed on November 5, 2015. Weather conditions were sunny with mild temperatures throughout the installation on both days.

6.1 Receiver Unit Installation & Analysis

The receiver unit installation contractor was Bruce & Merrilees. The contractor used a Ford F-450 bucket truck. The receiver unit installation and testing (See Section 7.2.2) required about 4 hours.

The installation requires a 6 inch (L) by 6 inch (W) by 16 feet (H) post as the foundation for mounting the receiver unit metal enclosure, the antenna, and the solar panel. A breakaway was incorporated into the base of the post. The pole to be used should be in compliance with Traffic Standard 8702E from Publication 111. The post was mounted within the 15-foot right-of-way from the road. A hole was dug for mounting the post. High strength concrete (4000 psi Sakrete bag concrete mix) was used to fill in the hole around the post. While the concrete cured, wooden supports (i.e., 2 inch by 4 inch) were tacked to the post and secured to the ground using metal posts. The wooden supports remained for about a week after the installation completion.

To install the receiver unit and metal enclosure, first, the metal enclosure’s support bracket was bolted to the post, followed by mounting the metal enclosure on its support bracket. The bottom edge of the metal enclosure was approximately 53 inches (4.42 feet) above ground. The antenna was bolted to the post at approximately 85 inches (7.08 feet) above ground (i.e., to the antenna’s base), facing downstream, with a wooden shim (i.e., a section of 2 inch x 4 inch) inserted towards the top-side of the antenna in order to angle the antenna slightly downward (i.e., because of its mounted height). The excess cable insulation of the solar panel was cut off to provide extra insulation and protection for the antenna’s cable. DAP® Auto/Marine 100% RTV Silicone Sealant was used to close off gaps in the insulation. The solar panel was bolted to the post approximately 131 inches (10.92 feet) above ground (i.e., to the solar panel’s mounting bracket). The solar panel was angled at approximately 45 degrees from the post’s face. Closer to 90 degrees would provide more sunlight, but would allow leaves and snow to accumulate on the solar panel surface; therefore, a compromise was made. The completed receiver unit installation with labeling is shown in
Figure 21: Additionally, Figure 35 in Appendix A shows the confirmed mounted heights.

Figure 21: Receiver unit installation showing all components of the installation, including: the mounting post, the antenna and shim, the solar panel, the lightening rod, and wooden supports.

The patch antenna’s gain is about 8 dBiC, which is the peak gain in the direction normal to the front face of the patch antenna. The 3 dB beamwidth is a popular metric to describe the directivity of the antenna. It is desirable to estimate when a float-out device will come within the field-of-view (i.e., covered by the radiation beam of the antenna) of the receiver unit system. Considering the 3 dB beamwidth, the lower boundary field-of-view of the receiver system, $D_{\text{min},3dB}$, exists about 76.39 inches away from the front face of the mounting post. Because the gain of the antenna is high, the 10 dB beamwidth is instructive provide an estimate of where the antenna’s coverage will have nearly unity gain. In actuality, the
gain will be -2 dB at the 10 dB beamwidth boundary, down from the antenna’s 8 dBiC peak gain. The reason for choosing a 10 dB beamwidth instead of an 8 dB beamwidth (i.e., which would yield the 0 dB gain boundary) is because the antenna’s datasheet naturally provides a reticle at 10 dB, which reduces introducing errors due to extrapolating the 8 dB point on a log scale that has no finer tick marks. Considering the 10 dB beamwidth, the lower boundary field-of-view of the receiver system, \( D_{\text{min},10dB} \), exists about 17.24 inches away from the front face of the mounting post. Figure 22 shows the geometry of the beamwidth considerations and the expected field-of-view lower bounds. In order to determine the field-of-view for a particular beamwidth, the following equation provides the minimum distance away from the front face of the mounting post (see Figure 22): \( D_{\text{min,XdB}} = \tan(90^\circ - \frac{\theta}{2} - \phi) \cdot (85\ \text{inch} + 4.25\ \text{inch}) + 1.5\ \text{inch} \), where \( X \) in \( D_{\text{min,XdB}} \) is the beamwidth (e.g., 3 dB, 10 dB, etc.). It is important to note that the antenna gain falls sharply off at higher beamwidths, which leads to increasingly poorly wireless reception as the beamwidth widens, leading to less meaningful \( D_{\text{min,XdB}} \) estimates.

![Diagram showing antenna mounting angle and its influence on when a float-out device comes into the receiver unit system's field-of-view.](image)

Figure 22: Illustration showing the antenna mounting angle and its influence on when a float-out device comes into the receiver unit system’s field-of-view.
Despite the boundary estimates, it is worthwhile to explain the result for why a float-out device transmission could be received while the float-out device was under the bridge and upstream from the antenna (see Table 3 in Section 7.2.2). The antenna gain is strongest in a narrow beamwidth normal to the antenna’s front face. The antenna gain falls off sharply as the beamwidth increases. However, because the antenna is mounted near the bridge itself, when a float-out device is near a null in the antenna’s radiation pattern (e.g., the 40 dB beamwidth, corresponding to $\theta \approx 180^\circ$), the float-out device’s proximity (i.e., low path loss) to the antenna along with its transmit power compensates for the poor antenna gain and results in a received transmission. Therefore, when the float-out devices are in close proximity to the antenna, regardless of being out of the field-of-view, it is possible for the float-out device to communicate with the receiver unit. However, communication in this region may not be consistent because the wireless channel is operating near the lower bound of the wireless power link budget (i.e., antenna gain, float-out device transmit power, path loss, etc.), which explains the results for tests under the bridge.

6.2 Float-out Device Installation & Analysis

The float-out locations were selected to monitor scour where it is likely to occur and where it already has occurred. The system can support up to sixteen (16) float-out device sensors. Figure 23 shows the locations for each of float-out installations that required monitoring. Significant scour was present at both location 1 and location 2. Due to flow conditions present at the bridge, it is expected that location 3 would also scour. Location 4 was selected in the middle of the stream, because it would provide a general scour case of the streambed.
Figure 23: Approximate installation locations for each of the four selected locations

L.G. Hetager Drilling, Inc. was the drilling contractor used to install the float-out devices. The drill rig was a Gardner Denver articulated drill rig with a track mount base, secured to lowboy trailer with geotechnical mast and tooling. The secured rig was backed into location near the guard rail on the upstream side of the bridge (see Figure 24). A wooden support board was placed from one wingwall to the other wingwall to provide a support for the drill rig and tooling (see Figure 25). The drill rig pumps water from the downstream side, to provide a lubrication during drilling. The articulating nature of the drill rig allowed for faster and more accurate alignment of the casing to the exact location for each drilled hole. In addition, the transition and setup time between each drilled hole can be reduced through prior planning and an understanding of the extension and rotation capabilities of the drill rig. The first and second locations were drilled without the need to move the lowboy trailer. However, the drilling of the third and fourth locations required re-positioning to be properly oriented for drilling. The setup and transition time required to move the trailer was approximately 20 minutes. Careful selection should be taken prior to the installation to select a location for the lowboy trailer to reduce the installation time. The float-out device holes were drilled using a 4-inch flush joint casing that could maintain a constant diameter hole.
Figure 24: Gardner Denver articulated drill rig track mount secured to a lowboy trailer

Figure 25: Articulating rig with geotechnical mast, tooling, and platform that are extended over the bridge and supported by a wood block
The drill time for location 1, location 2, and location 4 was approximately 45 minutes. It was believed a large boulder was encountered during the drilling of location 3, which took approximately 90 minutes to drill. The installation of each individual float-out device took approximately 15 minutes. The objective was to install four capsules with a spacing of 16 inches between each capsule. This configuration requires a total drilled depth of 8 feet. Once a depth of 8 feet has been drilled, the drill bit is removed from inside the casing and the casing is withdrawn by 1 foot from the bottom of the hole. At this point, water has flooded the inside of the well casing, and in certain cases, may be significantly higher than the water level outside of the stream. Immediately prior to lowering the float-out device into the casing, the receiver unit is checked to ensure that the float-out device has not been triggered. If the float-out device has not been triggered, then the float-out device is lowered into the casing, using the custom jig shown in Figure 26. The jig consists of a thin rope, knotted around a key ring, which is looped through a rubber band. Once the float-out device has reached the top of the water level inside of the casing, a quick jerk of the string will release the float-out device into the casing. While the use of the installation jig is not required, it was developed to reduce any possible stresses on the capsule during the installation. During the installation process, multiple float-out devices were released early from the jig and fell into the water inside the casing.

![Installation jig for the float-out device](image)

Figure 26: Installation jig for the float-out device

After the float-out device has been released into the casing, a custom PVC tamper was used to force the float-out device into the water inside the casing and to hold it in place in the sediment. The custom PVC tamper consisted of multiple 8 feet segments of 1 inch diameter PVC rods with threaded ends allowing for variable extension lengths. At the base
of the first section of PVC is a round solid PVC segment, approximately 3 inches in diameter, which acts as the tamping mechanism. With the float-out device secured by the tamping rod, sediment is added to the well casing to surround and lock the float-out device into position. Three types of material were used to create the backfill sediment: 20% fine sand (20/40 bagged sand, by Best Sand Corp.), 50% coarse sand (4/30 bagged sand, by Best Sand Corp.), and 30% mud from the streambed. The coarse sand mixed with the mud created a mixture similar to that naturally present in the stream. The fine sand helped to bind the coarse sand and mud together, making the backfill sediment mixture thicker, and easier to load into the 4 inch well casing. When the mixture combines with water, the fine sand will distribute and have little effect on the stability of the float-out device. Figure 27 shows measured bottom-edges of each installed float-out device. During the drilling of location 2 and location 4, the drill hole collapsed as the casing was withdrawn. The collapse is due to locations of soft sediment, which wash out as hole is drilled. The collapse required additional backfill to be added to the casing which height could only be estimated. Due to the estimation of backfill after the location 2 hole collapsed, only three of the four float-out devices were installed.
Figure 27: Installation depths of each installed float-out sensor.

During the installation, much effort was spent measuring the appropriate amount of sediment to backfill on top of the float-out device. Figure 28 shows the method used to determine the appropriate amount of sediment used to backfill between successive float-out devices. According to the figure, $H_{bucket, adj}$ is the height that backfill sediment should be filled to in the bucket, for a target backfill sediment height (i.e., $H_{backfill}$) About 1 foot of backfill sediment was the target between each float-out device. However, Figure 27 clearly shows that the amount of backfill sediment cannot be exactly determined. To partially combat this problem, during the installation, a correction factor ("C" in Figure 28) was applied to account for backfill sediment losses (i.e., due to case withdrawal, spillage, etc.).
Example

Using:

\[ Volume \ (\text{inch}^3) = \pi \left( \frac{\text{Diameter} \ (\text{inch})}{2} \right)^2 \cdot \text{Height} \ (\text{inch}) \]

Assume:

\[ D_{\text{bucket}} = 12\text{inch} \]
\[ H_{\text{backfill}} = 12\text{inch} \]

Then:

\[ V_{\text{bucket}}(\text{inch}^3) = \pi \cdot \left[ \left( \frac{4\text{inch}}{2} \right)^2 - \left( \frac{2\frac{1}{2}\text{inch}}{2} \right)^2 \right] \cdot \frac{7\frac{1}{8}\text{inch}}{2} + \pi \left( \frac{8\text{inch}}{2} \right)^2 \cdot \frac{V_2}{2} \]

\[ H_{\text{bucket}}(\text{inch}) = \frac{V_{\text{bucket}}}{\pi \left( \frac{D_{\text{bucket}} \ (\text{inch})}{2} \right)^2} \]

\[ H_{\text{bucket}}(\text{inch}) \approx 1.75\text{inch} \]
\[ H_{\text{bucket,adj}}(\text{inch}) = C \times 1.75\text{inch}, \text{where } C = 1.5 \]
\[ = 2.625\text{inch} \]

Figure 28: Method of determining amount of backfill sediment to provide a desired amount of backfill between successive float-out devices within the same scour hole. Here, 12 inches of backfill is the target.
7 Laboratory and Field Validation Testing

This section details the validation testing that was performed in the laboratory environment and at the installation site.

7.1 Laboratory Testing Receiver Unit and Float-out Devices

7.1.1 Receiver Unit Software Testing

The receiver unit required testing under various expected and unexpected float-out device transmission scenarios. The following test cases were performed using the receiver unit and a float-out device configured to transmit various scour hole numbers, scour color depth indicators, and both valid and invalid unique identification numbers. A valid unique identification number refers to an identification number currently loaded onto an installed float-out device; whereas, an invalid identification number is a number not programmed into any of the installed float-out devices. All of the following tests passed and therefore the receiver unit is approved for installation. Note that in the unlikely event that the LED matrix LEDs do not automatically turn off after four (4) seconds, simply press the pushbutton switch again; the LED matrix will automatically turn off afterward.

Table 2: Receiver Unit Validation Test Sequences.

<table>
<thead>
<tr>
<th>(Start Condition)</th>
<th>Begin testing after clearing stored scour locations (i.e., reset)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Trigger light indicator LEDs after transmission from float-out device</td>
</tr>
<tr>
<td>2.</td>
<td>Reset/clear stored scour locations</td>
</tr>
<tr>
<td>3.</td>
<td>Trigger light indicator LEDs during transmission from a float-out device</td>
</tr>
<tr>
<td>4.</td>
<td>Power cycle the receiver unit</td>
</tr>
<tr>
<td>5.</td>
<td>Trigger light indicator LEDs after transmission from a float-out device</td>
</tr>
<tr>
<td>6.</td>
<td>Power cycle the receiver unit</td>
</tr>
<tr>
<td>7.</td>
<td>Trigger light indicator LEDs after transmission from a float-out device</td>
</tr>
</tbody>
</table>

Transmit *one* valid identification number (Run test with (4) identification numbers to check basic operation)
<table>
<thead>
<tr>
<th></th>
<th>Trigger light indicator LEDs and hold pushbutton; when the reset sequence begins, trigger the float-out device.</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.</td>
<td>Trigger light indicator LEDs and hold pushbutton; trigger the float-out device eight (8) seconds into holding the pushbutton.</td>
</tr>
</tbody>
</table>

Transmit *multiple* valid identification numbers. All sixteen identification numbers will be repeatedly transmitted.

<table>
<thead>
<tr>
<th>(Start Condition)</th>
<th>Begin testing after clearing stored scour locations (i.e., reset)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Trigger light indicator LEDs <em>after</em> transmission from float-out</td>
</tr>
<tr>
<td>2.</td>
<td>Reset/clear stored scour locations</td>
</tr>
<tr>
<td>3.</td>
<td>Trigger light indicator LEDs <em>during</em> transmission from a float-out device</td>
</tr>
<tr>
<td>4.</td>
<td>Power cycle the receiver unit</td>
</tr>
<tr>
<td>5.</td>
<td>Trigger light indicator LEDs <em>after</em> transmission from a float-out device</td>
</tr>
<tr>
<td>6.</td>
<td>Power cycle the receiver unit</td>
</tr>
<tr>
<td>7.</td>
<td>Trigger light indicator LEDs <em>after</em> transmission from a float-out device</td>
</tr>
<tr>
<td>8.</td>
<td>Trigger light indicator LEDs and hold pushbutton; when the reset sequence begins, trigger the float-out device.</td>
</tr>
<tr>
<td>9.</td>
<td>Trigger light indicator LEDs and hold pushbutton; trigger the float-out device eight (8) seconds into holding the pushbutton.</td>
</tr>
</tbody>
</table>

Transmit *multiple* invalid identification numbers. All sixteen identification numbers will be repeatedly transmitted.

<table>
<thead>
<tr>
<th>(Start Condition)</th>
<th>Begin testing after clearing stored scour locations (i.e., reset)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Trigger light indicator LEDs <em>after</em> transmission from float-out</td>
</tr>
<tr>
<td>2.</td>
<td>Reset/clear stored scour locations</td>
</tr>
<tr>
<td>3.</td>
<td>Trigger light indicator LEDs <em>during</em> transmission from a float-out device</td>
</tr>
<tr>
<td>4.</td>
<td>Power cycle the receiver unit</td>
</tr>
<tr>
<td>5.</td>
<td>Trigger light indicator LEDs <em>after</em> transmission from a float-out device</td>
</tr>
<tr>
<td>6.</td>
<td>Power cycle the receiver unit</td>
</tr>
<tr>
<td></td>
<td>Trigger light indicator LEDs after transmission from a float-out device</td>
</tr>
<tr>
<td>---</td>
<td>------------------------------------------------------------------------</td>
</tr>
<tr>
<td>8.</td>
<td>Trigger light indicator LEDs and hold pushbutton; when the reset sequence begins, trigger the float-out device.</td>
</tr>
<tr>
<td>9.</td>
<td>Trigger light indicator LEDs and hold pushbutton; trigger the float-out device eight (8) seconds into holding the pushbutton.</td>
</tr>
</tbody>
</table>

Transmit a combination of valid and invalid identification numbers. All sixteen identification numbers will be repeatedly transmitted; half are valid, and half are invalid.

<table>
<thead>
<tr>
<th>(Start Condition)</th>
<th>Begin testing after clearing stored scour locations (i.e., reset)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Trigger light indicator LEDs after transmission from float-out device</td>
</tr>
<tr>
<td>2.</td>
<td>Reset/clear stored scour locations</td>
</tr>
<tr>
<td>3.</td>
<td>Trigger light indicator LEDs during transmission from a float-out device</td>
</tr>
<tr>
<td>4.</td>
<td>Power cycle the receiver unit</td>
</tr>
<tr>
<td>5.</td>
<td>Trigger light indicator LEDs after transmission from a float-out device</td>
</tr>
<tr>
<td>6.</td>
<td>Power cycle the receiver unit</td>
</tr>
<tr>
<td>7.</td>
<td>Trigger light indicator LEDs after transmission from a float-out device</td>
</tr>
<tr>
<td>8.</td>
<td>Trigger light indicator LEDs and hold pushbutton; when the reset sequence begins, trigger the float-out device.</td>
</tr>
<tr>
<td>9.</td>
<td>Trigger light indicator LEDs and hold pushbutton; trigger the float-out device eight (8) seconds into holding the pushbutton.</td>
</tr>
</tbody>
</table>

Transmit multiple invalid identification numbers, followed by transmitting multiple valid identification numbers. All sixteen valid and invalid identification numbers will be repeatedly transmitted.

<table>
<thead>
<tr>
<th>(Start Condition)</th>
<th>Begin testing after clearing stored scour locations (i.e., reset)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Trigger light indicator LEDs after transmission from the float-out device with invalid identification numbers.</td>
</tr>
<tr>
<td>2.</td>
<td>Trigger light indicator LEDs after transmission from the float-out device with valid identification numbers.</td>
</tr>
<tr>
<td>3.</td>
<td>Reset/clear stored scour locations</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>4.</td>
<td>Trigger light indicator LEDs <em>during</em> transmission from the float-out device with invalid identification numbers.</td>
</tr>
<tr>
<td>5.</td>
<td>Trigger light indicator LEDs <em>during</em> transmission from the float-out device with valid identification numbers.</td>
</tr>
<tr>
<td>6.</td>
<td>Power cycle the receiver unit</td>
</tr>
<tr>
<td>7.</td>
<td>Trigger light indicator LEDs <em>after</em> transmission from the float-out device with invalid identification numbers.</td>
</tr>
<tr>
<td>8.</td>
<td>Trigger light indicator LEDs <em>after</em> transmission from the float-out device with valid identification numbers.</td>
</tr>
<tr>
<td>9.</td>
<td>Power cycle the receiver unit</td>
</tr>
<tr>
<td>10.</td>
<td>Trigger light indicator LEDs <em>during</em> transmission from the float-out device with invalid identification numbers.</td>
</tr>
<tr>
<td>11.</td>
<td>Trigger light indicator LEDs <em>during</em> transmission from the float-out device with valid identification numbers.</td>
</tr>
<tr>
<td>12.</td>
<td>Trigger light indicator LEDs and hold pushbutton; when the reset sequence begins, trigger the float-out device with invalid identification numbers.</td>
</tr>
<tr>
<td>13.</td>
<td>Trigger light indicator LEDs and hold pushbutton; trigger the float-out device, with invalid identification numbers, eight (8) seconds into holding the pushbutton.</td>
</tr>
<tr>
<td>14.</td>
<td>Trigger light indicator LEDs and hold pushbutton; when the reset sequence begins, trigger the float-out device with valid identification numbers.</td>
</tr>
<tr>
<td>15.</td>
<td>Trigger light indicator LEDs and hold pushbutton; trigger the float-out device, with valid identification numbers, eight (8) seconds into holding the pushbutton.</td>
</tr>
</tbody>
</table>

### 7.1.2 Sealed Float-out Device Laboratory Testing

With the receiver unit installed in the metal enclosure, and powered by the battery controller system, the receiver unit is tested with the sealed float-out devices to ensure that all devices are operational before leaving the laboratory. Beginning with device #1 through #16 (see Figure 8), the testing protocol for the laboratory testing is straightforward for each float-out device under test (FDUT):
1. Remove the magnet superglued to the PVC end-cap (see Figure 29)

2. Trigger the float-out device under test (FDUT) once (i.e., tilt the device 90 degrees).

3. Wait until after the FDUT has finished transmitting (i.e., an audible click indicates that the electronic FDUT’s relay has disconnected its battery).

4. Observe the scour hole number (e.g., #1, #2, etc.) and scour color depth indicators (e.g., GREEN, YELLOW, etc.) via the light indicator LEDs on the receiver unit by pressing and quickly releasing the receiver unit’s pushbutton switch.

5. Confirm that the light indicator LED scour hole number and scour color depth indicator match that of the FDUT (i.e., painted number on the FDUT and painted color of the FDUT).

6. Clear/reset the receiver unit’s memory of FDUTs using the receiver unit’s pushbutton switch (see Section 5.1).

7. Superglue the magnet back onto the PVC end-cap in the exact spot it was removed.

![Image of magnet removal](image-url)

Figure 29: Removal of the magnet superglued to the PVC end-cap. Removing the magnet deactivates the anti-trigger feature, and permits the float-out device to trigger upon tilting the device.
Figure 30 shows a grid confirming that each of the sixteen (16) FDUTs actuated their respective light indicator LED.

![Grid showing the receiver unit registering each float-out device's wireless transmission signature.](image)

Figure 30: Grid showing the receiver unit registering each float-out device’s wireless transmission signature.

### 7.2 Field Installation Testing Receiver Unit and Float-out Devices

For information regarding the installation procedures, see Section 6.

#### 7.2.1 Field Testing the Float-out Devices

The goal of the float-out device testing was to ensure device operation prior to installation. Since the float-out devices were being installed before the receiver unit, the float-out device field test simply confirmed that each device was operational before finally installing each
into their scour holes. Therefore, the testing setup and testing protocol were simple.

**Float-out device Field Test Materials**

1. Receiver Unit mounted within its metal enclosure is positioned at end of a flat-bed pickup truck, with the antenna pointed towards the float-out device installation location.

2. A float-out device with magnet superglued to the PVC end-cap to prevent triggering.

**Float-out device Field Test Methodologies**

1. The magnet superglued to the PVC end-cap is removed. Hold the float-out device upright so as to not trigger it.

2. Tilt the float-out device 90 degrees to trigger it, then hold the device upright.

3. Press the pushbutton switch on the receiver unit to observe whether the scour location and scour depth, corresponding to the float-out device under test, has registered on the receiver unit LED matrix.

4. Wait at least 20 seconds for the float-out device timer to turn the float-out device off.

5. Clear the receiver unit memory. Ensure that no float-out devices have registered with the receiver unit.

The results of the float-out device field test confirmed that all float-out devices were operational before they were installed.

**7.2.2 Field Testing the Receiver Unit**

The goal of the receiver unit testing is two-fold: first, to verify the receiver unit is operational after installation, and second, to verify the proper positioning of the antenna. Note that the spare float-out device used during the receiver unit field test was device #5 (i.e., GREEN, LOC 2), which was not installed during the float-out device installation.

**Receiver Unit Field Test Materials**

1. Access to the receiver unit PCB within its metal enclosure.
2. Antenna pointed downstream.

3. Spare float-out device.

4. Waders to enter the stream.

5. Two (2) personnel: one to operate the receiver unit and one to trigger the spare float-out device in the stream.

Receiver Unit Field Testing Methodologies

1. A location within the stream is chosen as a test point.

2. The magnet superglued to the PVC end-cap is removed. Hold the float-out device upright so as to not trigger it.

3. As best as possible (i.e., stream depth permitting), submerge the float-out device without triggering it.

4. Release the float-out device allowing it to float downstream.

5. The operator at the receiver unit presses the pushbutton switch to observe whether the scour location and scour depth, corresponding to the float-out device used for testing, have registered on the receiver unit LED matrix.

6. Clear the receiver unit scour memory.

7. Retrieve the float-out device and superglue the magnet back onto the PVC end-cap.

Receiver Unit Field Test Results

Results of the receiver unit testing are found in Table 3. The results of the field test show that only after a deployed float-out device has passed the downstream wingwall will the receiver unit register a float-out device. The results confirmed that stream meandering did not impede the communication link between the deployed float-out device and receiver unit. Additionally, distance tests were limited by personnel access to the stream. The farthest test location, at approximately 486 feet from the receiver unit, was the farthest that could be tested before the stream bank height/steepness and surrounding brush impeded testing. Note that for the test case under the bridge, the Pass/Fail status depended on whether the float-out device was nearer the downstream side of the bridge or nearer the upstream, with the former resulting in a pass status, and the latter resulting in a fail status.
boundary for pass or fail under the bridge could not be precisely determined, due to antenna coverage which favored downstream as well as the bridge’s material properties (i.e., concrete and metal), which significantly impede radio transmissions. This result was expected and is a limitation of the used antenna type, which favored downstream distance and breadth coverage. In future installations, upstream coverage could be covered using an additional antenna pointed upstream.

Table 3: Receiver Unit Field Testing Results

<table>
<thead>
<tr>
<th>Test Location</th>
<th>Position relative to Antenna</th>
<th>Pass/Fail?</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Behind upstream wingwall</td>
<td>upstream</td>
<td>Fail</td>
<td>Same location as float-out installation scour holes</td>
</tr>
<tr>
<td>Under the bridge</td>
<td>upstream side</td>
<td>Fail</td>
<td>N/A</td>
</tr>
<tr>
<td>Under the bridge</td>
<td>downstream side</td>
<td>Pass</td>
<td>N/A</td>
</tr>
<tr>
<td>About six (6) feet forward the bridge</td>
<td>downstream</td>
<td>Pass</td>
<td>See Figure 38</td>
</tr>
<tr>
<td>At first stream bend</td>
<td>downstream</td>
<td>Pass</td>
<td>See Figure 37</td>
</tr>
<tr>
<td>Estimated at 162 paces (≈486 feet) for</td>
<td>downstream</td>
<td>Pass</td>
<td>See Figure 36. Note that the figure shows approximate location of the downstream operator.</td>
</tr>
</tbody>
</table>
A Receiver Unit Validation Images

Receiver Unit Hardware
The figures below refer to Section 3.2.2. Shown are the two delivered Receiver Unit PCBs in Figure 31, along with their associated PCB layout artwork in Figure 32 and schematic in Figure 33. Also shown below is the receiver unit current consumption, about 25.15 mA (i.e., 0.02515 A), at 12 V in Figure 34. Referring to Section 3.2.1, this standby current consumption will provide 69 days of autonomy from the battery.

Figure 31: The two manufactured and assembled receiver unit printed circuit boards.
Figure 32: Receiver unit printed circuit board layout in ExpressPCB.
Figure 33: Receiver unit printed circuit board schematic in ExpressPCB.
Figure 34: Receiver unit current consumption from a 12 V supply. Current consumption is about 25.15 mA (i.e., 0.02515 A), which equals about 0.3018 W. The receiver will be autonomous for about 69 days at this power consumption.
**Receiver Unit Field Test Results**

The figures below refer to Section 7.2.2. Shown are the mounted heights of the receiver unit enclosure, antenna, and solar panel in Figure 35. Also shown are field test images where float-out device sensors were tested in the stream at various distances away from the receiver unit (Figures 36, 37, and 38).

Figure 35: Mounting heights of the receiver unit enclosure, antenna, and solar panel.

Figure 36: Float-out device tested about 486 feet (162 paces) downstream.
Figure 37: Float-out device tested at the first stream bend.

Figure 38: Float-out device tested about six (6) feet in downstream forward the bridge.
B  Float-out Device Validation Images

The figures below refer to Section 4. Shown below are the sixteen (16) float-out device electronics in Figure 39 and then the sealed, painted, and marked float-out devices in Figure 40. In Figure 40, the color and numbers correspond to the scour color depth indicator and scour hole number, respectively, of the specific float-out device. See Section 4.1 for further explanation.

Figure 39: The sixteen (16) float-out device PCBs that were sealed into the capsules shown in Figure 40.

Figure 40: The sixteen (16) sealed, painted, and marked float-out devices, showing front and reverse sides.
C Appendix: Receiver Unit Bill of Materials

Figure 33 from Appendix A shows the receiver unit, which corresponds to the PCB layout in Figure 32. The Bill of Materials “schematic label” corresponds to components in the schematic. Additionally, the majority of components are available from the distributor website DigiKey.com; however, if a part is only available from another supplier, and not DigiKey.com, then this cell lists the website where the part was sourced along with a part number.

<table>
<thead>
<tr>
<th>Schematic Label</th>
<th>Qty</th>
<th>Description</th>
<th>DigiKey Part Number</th>
<th>Manufacturer Part Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>U7–U14</td>
<td>8</td>
<td>TRANS 2NPN 40V 0.5A SC70-6</td>
<td>FFB2222ACT-ND</td>
<td>FFB2222A</td>
</tr>
<tr>
<td>D1, D5, D9, D13</td>
<td>4</td>
<td>LED GREEN CLEAR 5MM ROUND T/H</td>
<td>365-1181-ND</td>
<td>OVLFG3C7</td>
</tr>
<tr>
<td>D2, D6, D10, D14</td>
<td>4</td>
<td>LED YELLOW CLEAR 5MM ROUND T/H</td>
<td>67-1115-ND</td>
<td>SSL-LX5093SYC</td>
</tr>
<tr>
<td>D4, D8, D12, D16</td>
<td>4</td>
<td>LED ORANGE CLEAR 5MM ROUND T/H</td>
<td>67-1113-ND</td>
<td>SSL-LX5093SOC</td>
</tr>
<tr>
<td>D3, D7, D11, D15</td>
<td>4</td>
<td>LED RED CLEAR 5MM ROUND T/H</td>
<td>365-1182-ND</td>
<td>OVLFR3C7</td>
</tr>
<tr>
<td>U2</td>
<td>1</td>
<td>IC I/O EXPANDER SPI 16B 28SSOP</td>
<td>MCP23S17T-E/SSCT-ND</td>
<td>MCP23S17T-E/SS</td>
</tr>
<tr>
<td>U3</td>
<td>1</td>
<td>IC LVL XLTR LV 8MBPS 14-TSSOP</td>
<td>MAX3378EEUD+TCT-ND</td>
<td>MAX3378EEUD+T</td>
</tr>
<tr>
<td>U4</td>
<td>1</td>
<td>IC SOC RF TXRX W/8051 MCU 36-VQF</td>
<td>296-38889-1-ND</td>
<td>CC1110F32RHHT</td>
</tr>
<tr>
<td>FL1</td>
<td>1</td>
<td>FILTER BANDPASS 902-928MHZ TI</td>
<td>712-1541-1-ND</td>
<td>0915BM15A0001E</td>
</tr>
<tr>
<td>U5</td>
<td>1</td>
<td>IC REG LDO 5V 0.75A TO252-5</td>
<td>296-27708-1-ND</td>
<td>TL751M05QKVURQ1</td>
</tr>
<tr>
<td>U6</td>
<td>1</td>
<td>IC REG LDO 3V 0.15A SOT23-5</td>
<td>296-35482-1-ND</td>
<td>TPS70930DBVR</td>
</tr>
<tr>
<td>Q1</td>
<td>1</td>
<td>MOSFET N-CH 30V 200MA TO-236AB</td>
<td>568-10504-1-ND</td>
<td>NX3020NAK,215</td>
</tr>
<tr>
<td>X1</td>
<td>1</td>
<td>Crystal 26.0000MHz 20ppm 18pF 60 Ohm -40°C – 85°C Surface Mount 4-SMD, No Lead (DFN, LCC)</td>
<td>535-10276-1-ND</td>
<td>ABM8G-26.000MHZ-18-D2Y-T</td>
</tr>
<tr>
<td>SW1</td>
<td>1</td>
<td>SWITCH PUSH SPST-NO 0.5A 125V</td>
<td>450-1098-ND</td>
<td>MSPF101C</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td><strong>J1</strong></td>
<td>1</td>
<td>CONN SMA JACK STR 50 OHM PCB</td>
<td>A97594-ND</td>
<td>5-1814832-1</td>
</tr>
<tr>
<td><strong>J2</strong></td>
<td>1</td>
<td>CONN HEADER .050&quot; 10PS DL SMD AU</td>
<td>S9012E-05-ND</td>
<td>GRPB052VWQS-RC</td>
</tr>
<tr>
<td><strong>J3</strong></td>
<td>1</td>
<td>Connector Barrier Block Strip 2 Circuit 0.375&quot; (9.53mm)</td>
<td>1437664-4-ND</td>
<td>1437664-4</td>
</tr>
<tr>
<td><strong>R28–R43</strong></td>
<td>16</td>
<td>RES SMD 150 OHM 1% 1/8W 0805</td>
<td>1276-5230-1-ND</td>
<td>RC2012F151CS</td>
</tr>
<tr>
<td><strong>R10–R25</strong></td>
<td>16</td>
<td>RES SMD 1K OHM 1% 1/8W 0805</td>
<td>P1.00KCCT-ND</td>
<td>ERJ-6ENF1001V</td>
</tr>
<tr>
<td><strong>R9</strong></td>
<td>1</td>
<td>RES SMD 2.7K OHM 1% 1/16W 0402</td>
<td>RMCF0402FT2K70CT-ND</td>
<td>RMCF0402FT2K70</td>
</tr>
<tr>
<td><strong>R2</strong></td>
<td>1</td>
<td>RES SMD 39K OHM 5% 1/16W 0402</td>
<td>311-39KJRCT-ND</td>
<td>RC0402JR-0739KL</td>
</tr>
<tr>
<td><strong>R44</strong></td>
<td>1</td>
<td>RES SMD 100K OHM 1% 1/16W 0402</td>
<td>311-100KLRCT-ND</td>
<td>RC0402FR-07100KL</td>
</tr>
<tr>
<td><strong>R4</strong></td>
<td>1</td>
<td>RES SMD 3K OHM 5% 1/16W 0402</td>
<td>311-3.0KJRCT-ND</td>
<td>RC0402JR-073KL</td>
</tr>
<tr>
<td><strong>R5</strong></td>
<td>1</td>
<td>RES SMD 100 OHM 5% 1/16W 0402</td>
<td>311-100JRCT-ND</td>
<td>RC0402JR-07100RL</td>
</tr>
<tr>
<td><strong>R8</strong></td>
<td>1</td>
<td>RES SMD 56K OHM 0.1% 1/16W 0402</td>
<td>P56KDCCT-ND</td>
<td>ERA-2AEB563X</td>
</tr>
<tr>
<td><strong>R27</strong></td>
<td>1</td>
<td>RES SMD 220K OHM 5% 1/16W 0402</td>
<td>311-220KJRCT-ND</td>
<td>RC0402JR-07220KL</td>
</tr>
<tr>
<td><strong>R26</strong></td>
<td>1</td>
<td>RES SMD 62K OHM 5% 1/16W 0402</td>
<td>311-62KJRCT-ND</td>
<td>RC0402JR-0762KL</td>
</tr>
<tr>
<td><strong>C24</strong></td>
<td>1</td>
<td>CAP TANT 10UF 10V 10% 1210</td>
<td>478-1672-1-ND</td>
<td>TAJB106K010RNJ</td>
</tr>
<tr>
<td><strong>C9, C16</strong></td>
<td>2</td>
<td>CAP CER 30PF 50V N0 0402</td>
<td>490-8198-1-ND</td>
<td>GRM1555C1H300GA01D</td>
</tr>
<tr>
<td><strong>C12</strong></td>
<td>1</td>
<td>CAP CER 0.022UF 16V X7R 0402</td>
<td>709-1128-1-ND</td>
<td>160R.07W223KV4T</td>
</tr>
<tr>
<td><strong>C3, C4, C6, C8, C10, C11, C13, C15, C17–C19, C21</strong></td>
<td>12</td>
<td>CAP CER 0.1UF 25V X5R 0402</td>
<td>490-5920-1-ND</td>
<td>GRM155R61E104KA87D</td>
</tr>
<tr>
<td><strong>C2, C5, C7, C20</strong></td>
<td>4</td>
<td>CAP CER 1UF 35V X5R 0402</td>
<td>445-9073-1-ND</td>
<td>C1005X5R1V105K050BC</td>
</tr>
<tr>
<td><strong>C14</strong></td>
<td>1</td>
<td>CAP CER 1000PF 50V N0 0402</td>
<td>490-6189-1-ND</td>
<td>GRM1555C1H102FA01D</td>
</tr>
<tr>
<td>Part/Description</td>
<td>Quantity</td>
<td>Type/Model</td>
<td>Part Number</td>
<td>Supplier</td>
</tr>
<tr>
<td>------------------</td>
<td>----------</td>
<td>------------</td>
<td>-------------</td>
<td>----------</td>
</tr>
<tr>
<td>C25, C26</td>
<td>2</td>
<td>CAP CER 4.7UF 25V X7R 0805</td>
<td>490-9962-1-ND</td>
<td>GRM21BR71E475KA73L</td>
</tr>
<tr>
<td>C1, C22</td>
<td>1</td>
<td>CAP CER 22UF 35V X5R 1206 1206</td>
<td>445-8045-1-ND</td>
<td>C3216X5R1V226M160AC</td>
</tr>
<tr>
<td>C23</td>
<td>1</td>
<td>CAP CER 1UF 50V X7R 0805</td>
<td>490-4736-1-ND</td>
<td>GRM21BR71H105KA12L</td>
</tr>
<tr>
<td>FB1</td>
<td>1</td>
<td>FERRITE BEAD 120 OHM 3.0A 1206</td>
<td>732-1622-1-ND</td>
<td>742792113</td>
</tr>
<tr>
<td>R6, R7</td>
<td>2</td>
<td>RES SMD 24K OHM 0.5% 1/16W 0402</td>
<td>P24KDECT-ND</td>
<td>ERA-2AED243X</td>
</tr>
</tbody>
</table>

### Additional Receiver Enclosure Parts

<table>
<thead>
<tr>
<th>Accessory</th>
<th>Description</th>
<th>Quantity</th>
<th>Part Number</th>
<th>Supplier</th>
</tr>
</thead>
<tbody>
<tr>
<td>SW1</td>
<td>BOOTSEAL BLK 15/32-32 MPG/MSPF</td>
<td>1</td>
<td>450-1554-ND</td>
<td>BP1532004</td>
</tr>
<tr>
<td>J1</td>
<td>CONN ADAPT JACK-JACK SMA 50 OHM</td>
<td>1</td>
<td>ACX1242-ND</td>
<td>132169</td>
</tr>
<tr>
<td>J1</td>
<td>FITTING CABLE PG11 THREAD METAL</td>
<td>1</td>
<td>A111823-ND</td>
<td>1-1106006-9</td>
</tr>
<tr>
<td>J1</td>
<td>CABLE SMA/SMA 36&quot; RG-58</td>
<td>1</td>
<td>J4636-ND</td>
<td>415-0038-036</td>
</tr>
<tr>
<td>J3</td>
<td>CONN RING 16-22 AWG #6 PIDG</td>
<td>2</td>
<td>A27292-ND</td>
<td>53406-1</td>
</tr>
<tr>
<td>PCB</td>
<td>SparkFun Projects Case - Clear</td>
<td>1</td>
<td>Sparkfun.com: WIG-08632</td>
<td>5KADHC</td>
</tr>
<tr>
<td>J1</td>
<td>900 MHz 8 dBi RH Circular Polarized Patch Antenna—4ft SMA Male Connector</td>
<td>1</td>
<td>L-com.com: HG908PCR-SM</td>
<td>422B-340G</td>
</tr>
<tr>
<td>J1</td>
<td>2 in. Laminated Steel Padlock</td>
<td>1</td>
<td>homedepot.com: 100112589</td>
<td>5KADHC</td>
</tr>
<tr>
<td>PCB</td>
<td>Chemicals CONFORMAL COATING 340G (12 OZ) AEROSOL</td>
<td>1</td>
<td>Mouser.com: 590-422B-340G</td>
<td>00694</td>
</tr>
<tr>
<td>PCB</td>
<td>Auto/Marine 100% RTV Silicone Sealant</td>
<td>1</td>
<td><a href="http://www.homedepot.com">www.homedepot.com</a>: 00694</td>
<td>00694</td>
</tr>
<tr>
<td>PCB</td>
<td>GROMMET 0.252&quot; ELASTOMER BLACK</td>
<td>4</td>
<td>RP454-ND</td>
<td>HG-4</td>
</tr>
<tr>
<td>Sparkfun Projects Case Accessory (need 4 screws)</td>
<td>2</td>
<td>1/4 inch-20 x 3-1/2 inch Phillips Flat-Head Machine Screws (2-Pack)</td>
<td><a href="http://www.homedepot.com">www.homedepot.com</a>: 44781</td>
<td>–</td>
</tr>
<tr>
<td>Sparkfun Projects Case Accessory (need 12 nuts)</td>
<td>3</td>
<td>1/4 inch -20 Stainless Steel Coarse Hex Nuts (4 per Pack)</td>
<td><a href="http://www.homedepot.com">www.homedepot.com</a>: 31901</td>
<td>–</td>
</tr>
<tr>
<td>6 inch x 6 inch x 16 feet Pole Mounting Accessory</td>
<td>–</td>
<td>high strength 4000 psi Sakrete bag concrete mix</td>
<td><a href="http://www.sakrete.com/">www.sakrete.com/</a></td>
<td>–</td>
</tr>
<tr>
<td>–</td>
<td>1</td>
<td>Power Ready System with, SPM020P-F module 20Wp Solar Array, pole mounted on 2-4(S)&quot; Side of Pole mount with no accessories. The 12Vdc, 126AH VRLA-GEL battery bank w/ surge arrestor, charge controller, and breakers are mounted in C1 MET ALU NEMA3R 17X16X9 w/ mil-finish aluminum enclosure</td>
<td>sunwizepower.com: PR-20-12-126-CPAA-111</td>
<td>–</td>
</tr>
</tbody>
</table>
D Appendix: Float-Out Device Bill of Materials, PCB, & Schematic

Figure 41 shows the float-out device PCB layout. Figure 42 is the float-out device schematic corresponding to the layout. The Bill of Materials ‘schematic label’ corresponds to components in the schematic. Additionally, the majority of components are available from the distributor website DigiKey.com; however, if a part is only available from another supplier, and not DigiKey.com, then this cell lists the website where the part was sourced along with a part number.

![Float-out device sensor PCB layout](image)

Figure 41: Float-out device sensor PCB layout. Use for reference with the schematic.
Figure 42: Schematic of the float-out device sensor. Use for reference with the Bill of Materials.
<table>
<thead>
<tr>
<th>Schematic Label</th>
<th>Qty</th>
<th>Description</th>
<th>DigiKey Part Number</th>
<th>Manufacturer Part Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>1</td>
<td>Cylindrical Battery Contacts, Clips, Holders &amp; Springs 1/2AA WIRES BLACK</td>
<td>Mouser.com: 12BH1/2AA-2A-GR</td>
<td>12BH1/2AA-2A-GR</td>
</tr>
<tr>
<td>A1</td>
<td>1</td>
<td>High Power Primary Li Metal Oxide Cell 1/2AA</td>
<td>evssupply.com: TLM 1530/HE/T</td>
<td>TLM-1530HE/T</td>
</tr>
<tr>
<td>J1</td>
<td>1</td>
<td>CONN RP-SMA JACK STR 50 OHM PCB</td>
<td>CONREV SMA001-ND</td>
<td>CONREV SMA001</td>
</tr>
<tr>
<td>AE1</td>
<td>1</td>
<td>Linx Antenna 916 MHz 1/2 Wave Dipole</td>
<td>ANT-916-CW-HW-ND</td>
<td>ANT-916-CW-HW</td>
</tr>
<tr>
<td>A2</td>
<td>1</td>
<td>BOARD SHIELD .65X.65&quot; FRAME</td>
<td>903-1051-1-ND</td>
<td>BMI-S-202-F</td>
</tr>
<tr>
<td>A2 accessory</td>
<td>1</td>
<td>BOARD SHIELD .65X.65&quot; COVER</td>
<td>903-1014-ND</td>
<td>BMI-S-202-C</td>
</tr>
<tr>
<td>FL1</td>
<td>1</td>
<td>FILTER SAW 915MHZ REMOTE SMD</td>
<td>495-1674-1-ND</td>
<td>B39921B3588U410W9</td>
</tr>
<tr>
<td>FL2</td>
<td>1</td>
<td>FILTER BANDPASS 902-928MHZ TI</td>
<td>712-1541-1-ND</td>
<td>0915B15A0001E</td>
</tr>
<tr>
<td>J2</td>
<td>1</td>
<td>CONN HEADER .050&quot; 10PS DL SMD AU</td>
<td>S9012E-05-ND</td>
<td>GRPB052VWQS-RC</td>
</tr>
<tr>
<td>SW2, SW3</td>
<td>2</td>
<td>TILT SWITCH</td>
<td>EG4917CT-ND</td>
<td>TM1000Q</td>
</tr>
<tr>
<td>SW1</td>
<td>1</td>
<td>SENSOR MAGNETIC SPST-NC .5A 175V</td>
<td>59065-040-ND</td>
<td>59065-040</td>
</tr>
<tr>
<td>SW1 accessory</td>
<td>1</td>
<td>MAGNET THRDDED BARREL 5/16-24UNF</td>
<td>57065-000-ND</td>
<td>57065-000</td>
</tr>
<tr>
<td>K1</td>
<td>1</td>
<td>RELAY GENERAL PURPOSE SPST 8A 3V</td>
<td>255-3541-ND</td>
<td>DSP1A-L2-DC3V</td>
</tr>
<tr>
<td>U5, U8</td>
<td>2</td>
<td>IC REG LDO 3V 0.15A 6SON</td>
<td>296-24636-1-ND</td>
<td>TPS78230DRVT</td>
</tr>
<tr>
<td>U7</td>
<td>1</td>
<td>IC REG LDO 3V 0.5A SOT223-6</td>
<td>296-13809-2-ND</td>
<td>TPS79530DCQR</td>
</tr>
<tr>
<td>Q1</td>
<td>1</td>
<td>TRANS NPN 40V 1A SOT-23</td>
<td>MMBT2222AFSCT-ND</td>
<td>MMBT2222A</td>
</tr>
<tr>
<td>X1</td>
<td>1</td>
<td>Crystal 26.0000MHz 10ppm 9pF 60 Ohm -20°C – 75°C Surface Mount 4-SMD, No Lead (DFN, LCC)</td>
<td>SER3679CT-ND</td>
<td>FA-128 26.0000MF10Z-AC3</td>
</tr>
<tr>
<td>Ref.</td>
<td>Value</td>
<td>Type</td>
<td>Qty.</td>
<td>Part Number</td>
</tr>
<tr>
<td>------</td>
<td>-------</td>
<td>------</td>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>C8, C10, C11, C13, C15, C17–C19</td>
<td>CAP CER 0.1UF 25V X5R 0402</td>
<td>8</td>
<td></td>
<td>490-5920-1-ND</td>
</tr>
<tr>
<td>R6, R7</td>
<td>RES SMD 10K OHM 1% 1/16W 0402</td>
<td>2</td>
<td></td>
<td>YAG1233CT-ND</td>
</tr>
<tr>
<td>R8</td>
<td>RES SMD 56K OHM 0.1% 1/16W 0402</td>
<td>1</td>
<td></td>
<td>P56KDCCCT-ND</td>
</tr>
<tr>
<td>C20</td>
<td>CAP CER 1UF 35V X5R 0402</td>
<td>1</td>
<td></td>
<td>445-9073-1-ND</td>
</tr>
<tr>
<td>C14</td>
<td>CAP CER 1000PF 50V NP0 0402</td>
<td>1</td>
<td></td>
<td>490-6189-1-ND</td>
</tr>
<tr>
<td>R9</td>
<td>RES SMD 2.7K OHM 1% 1/16W 0402</td>
<td>1</td>
<td></td>
<td>RMCF0402FT2K70CT-ND</td>
</tr>
<tr>
<td>C9, C16</td>
<td>CAP CER 8PF 50V C0G 0402</td>
<td>2</td>
<td></td>
<td>445-4890-1-ND</td>
</tr>
<tr>
<td>U4</td>
<td>IC SOC RF TXRX W/8051 MCU 36-VQF</td>
<td>1</td>
<td></td>
<td>290-38889-1-ND</td>
</tr>
<tr>
<td>C7, C27, C29, C30</td>
<td>CAP CER 4.7UF 16V X5R 0603</td>
<td>4</td>
<td></td>
<td>445-7478-1-ND</td>
</tr>
<tr>
<td>FB2, FB4, FB5</td>
<td>FERRITE CHIP 1000 OHM 300MA 0603</td>
<td>3</td>
<td></td>
<td>445-8672-1-ND</td>
</tr>
<tr>
<td>FB1</td>
<td>FERRITE CHIP 47 OHM 500MA 0603</td>
<td>1</td>
<td></td>
<td>445-8678-1-ND</td>
</tr>
<tr>
<td>R10</td>
<td>RES SMD 100 OHM 1% 1/10W 0402</td>
<td>1</td>
<td></td>
<td>MCS0402-100-CFCT-ND</td>
</tr>
<tr>
<td>FB3</td>
<td>FERRITE CHIP POWER 160 OHM SMD</td>
<td>1</td>
<td></td>
<td>240-2411-1-ND</td>
</tr>
<tr>
<td>C26, C31</td>
<td>CAP CER 22UF 6.3V X5R 0805</td>
<td>1</td>
<td></td>
<td>490-1719-1-ND</td>
</tr>
<tr>
<td>C28</td>
<td>CAP CER 0.1UF 50V X7R 0805</td>
<td>1</td>
<td></td>
<td>490-10782-1-ND</td>
</tr>
<tr>
<td>L1</td>
<td>FIXED IND 11NH 500MA 140 MOHM</td>
<td>1</td>
<td></td>
<td>490-6766-1-ND</td>
</tr>
<tr>
<td>C6</td>
<td>CAP CER 100UF 10V X5R 1206</td>
<td>1</td>
<td></td>
<td>445-6007-1-ND</td>
</tr>
<tr>
<td>R4, R5</td>
<td>RES SMD 10K OHM 1% 1/16W 0402</td>
<td>2</td>
<td></td>
<td>YAG2104CT-ND</td>
</tr>
<tr>
<td>R3</td>
<td>RES SMD 10 OHM 1% 1/6W 0402</td>
<td>1</td>
<td></td>
<td>408-1413-1-ND</td>
</tr>
<tr>
<td>R2</td>
<td>RES SMD 47 OHM 1% 1/6W 0402</td>
<td>1</td>
<td></td>
<td>408-1419-1-ND</td>
</tr>
<tr>
<td>R1</td>
<td>RES SMD 3.3K OHM 0.1% 1/16W 0402</td>
<td>1</td>
<td></td>
<td>A102835CT-ND</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>L2</td>
<td>1</td>
<td>FIXED IND 1.5NH 300MA 100 MOHM</td>
<td>490-2612-1-ND</td>
<td>LQG15HS1N5S02D</td>
</tr>
<tr>
<td>C1</td>
<td>1</td>
<td>CAP CER 15PF 50V NP0 0402</td>
<td>490-5888-1-ND</td>
<td>GRM1555C1H150JA01D</td>
</tr>
<tr>
<td>C3</td>
<td>1</td>
<td>CAP CER 27PF 50V NP0 0402 NP0</td>
<td>490-5869-1-ND</td>
<td>GRM1555C1H270JA01D</td>
</tr>
<tr>
<td>C4, C25</td>
<td>2</td>
<td>CAP CER 12PF 50V NP0 0402</td>
<td>490-5924-1-ND</td>
<td>GRM1555C1H120JA01D</td>
</tr>
<tr>
<td>C2, C12, C24</td>
<td>1</td>
<td>CAP CER 47PF 50V NP0 0402</td>
<td>720-1291-1-ND</td>
<td>VJ0402D470JXAAJ</td>
</tr>
<tr>
<td>L6</td>
<td>1</td>
<td>FIXED IND 7.5NH 570MA 130 MOHM</td>
<td>490-6846-1-ND</td>
<td>LQW15AN7N5G00D</td>
</tr>
<tr>
<td>C22</td>
<td>1</td>
<td>CAP CER 3.3PF 50V NP0 0402</td>
<td>399-1007-1-ND</td>
<td>C0402C339C5GACTU</td>
</tr>
<tr>
<td>L4</td>
<td>1</td>
<td>FIXED IND 2.9NH 750MA 70 MOHM</td>
<td>490-6801-1-ND</td>
<td>LQW15AN2N9B00D</td>
</tr>
<tr>
<td>C23</td>
<td>1</td>
<td>CAP CER 7.5PF 50V NP0 0402</td>
<td>490-6075-1-ND</td>
<td>GJM1555C1H7R5DB01D</td>
</tr>
<tr>
<td>L5</td>
<td>1</td>
<td>FIXED IND 9.1NH 540MA 140 MOHM</td>
<td>490-6854-1-ND</td>
<td>LQW15AN9N1G00D</td>
</tr>
<tr>
<td>L3</td>
<td>1</td>
<td>FIXED IND 22NH 640MA 99 MOHM SMD</td>
<td>490-6876-1-ND</td>
<td>LQW18AN22NJ10D</td>
</tr>
<tr>
<td>C5</td>
<td>1</td>
<td>CAP CER 1000PF 50V NP0 0402</td>
<td>399-8940-1-ND</td>
<td>C0402C102F5GACTU</td>
</tr>
<tr>
<td>C21</td>
<td>1</td>
<td>CAP CER 1UF 6.3V 10% X5R 0402</td>
<td>490-1320-1-ND</td>
<td>GRM155R60J105KE19D</td>
</tr>
<tr>
<td>U2</td>
<td>1</td>
<td>IC RF FRONT-END 16VQFN</td>
<td>296-25826-1-ND</td>
<td>CC1190RGVT</td>
</tr>
</tbody>
</table>

### Additional Float-out Sensor Device Parts

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Accessory</td>
<td>1</td>
<td>Auto/Marine 100% RTV Silicone Sealant</td>
<td><a href="http://www.homedepot.com">www.homedepot.com</a>: 00694</td>
<td>00694</td>
<td></td>
</tr>
<tr>
<td>Accessory</td>
<td>1</td>
<td>3.7 fl. oz. Amazing Goop. All-Purpose Adhesive</td>
<td><a href="http://www.homedepot.com">www.homedepot.com</a>: 140211</td>
<td>1000021</td>
<td></td>
</tr>
<tr>
<td>Casing</td>
<td>1</td>
<td>Standard-Wall White PVC Un-threaded Pipe 2-1/2 Pipe Size X 10' Length (makes 16)</td>
<td>McMaster-Carr.com: 48925K19</td>
<td>–</td>
<td></td>
</tr>
<tr>
<td>Accessory</td>
<td>1</td>
<td>Oatey 8 oz. PVC and CPVC Purple Primer</td>
<td><a href="http://www.homedepot.com">www.homedepot.com</a>: 3075633</td>
<td>–</td>
<td></td>
</tr>
<tr>
<td>End-caps</td>
<td>2</td>
<td>Chemical-Resistant PVC (Type I) Rod 2-7/8&quot; Diameter, 4 feet Length (makes 32)</td>
<td>McMaster-Carr.com: 8745K54</td>
<td>–</td>
<td></td>
</tr>
<tr>
<td>Accessory for PN: 57065-000-ND</td>
<td>1</td>
<td>Krazy Glue Instant Glue, Advanced Formula 0.18 fl oz (5 g)</td>
<td><a href="http://www.riteaid.com/0382944">www.riteaid.com/0382944</a></td>
<td>—</td>
<td></td>
</tr>
<tr>
<td>Accessory</td>
<td>1</td>
<td>JET 24” Parallel Clamp</td>
<td><a href="http://www.acmetools.com:E9FA27B4E8E">www.acmetools.com:E9FA27B4E8E</a></td>
<td>70424</td>
<td></td>
</tr>
<tr>
<td>Accessory</td>
<td>1</td>
<td>Anti-Static Foam - 1-Qtr Inches Thick 4 x 5 Inches</td>
<td><a href="http://www.elexp.com:0302ASF">www.elexp.com:0302ASF</a></td>
<td>—</td>
<td></td>
</tr>
<tr>
<td>Accessory</td>
<td>2</td>
<td>4/30 Bagged Sand</td>
<td>Best Sand Corporation</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td>Accessory</td>
<td>1</td>
<td>20/40 Bagged Sand</td>
<td>Best Sand Corporation</td>
<td>—</td>
<td></td>
</tr>
</tbody>
</table>
E  Appendix: Receiver Unit Embedded Code

The receiver unit code on-board the receiver units’ MCU requires three (3) primary files. The c-file scour_receiv er_c_phase3.c contains the program’s *main*; The scour_receiv er_h_phase3.h file is the accompanying header file. The flash_asm.s51 file was provided by Texas Instruments and must be included in the project.

The bridge identification number (i.e., decimal 3102802400395; 0x000002D26D6F548B) is stored in BrID_7 through BrID_0, where BrID_7 is the most significant byte. The bridge identification number in the receiver unit code is the same across all float-out devices in this Phase 3 installation.

```c
// INCLUDES

#include <scour_Receiver.h>

///// FLASH

// One whole page of flash memory is to be reserved, i.e., 1 kb
#define PAGE_SIZE 1024

// The area in flash where the string (data written to flash) will be placed.
Page 17 -> 0x4400 ( = 1024 * 17).
__no_init const char _code flashDataAddr[PAGE_SIZE] @ 0x4400;
__no_init const char _code flashDataAddr[PAGE_SIZE] @ 0x7800;

// Length of data to be written to flash memory
#define FLASH_DATA_LEN 2

// Storage array for temporary port information storage prior to flash write
static __xdata uint8 write_flash_data[FLASH_DATA_LEN] = {0x00, 0x00};

// Storage array for temporary port information storage prior to flash read
static __xdata uint8 read_flash_data[FLASH_DATA_LEN] = {0x00, 0x00};

///// END FLASH
```
* CONSTANTS

#define UART_BAUD_M 34  
#define UART_BAUD_E 11

// SPI baud rate = 4.01171875 MHz
#define SPI_BAUD_M 60  
#define SPI_BAUD_E 17

#define BUFFER_SIZE 0x05

#define SLAVE_ADDR_PREAMBLE 1  
#define SLAVE_ADDR_EXT 0  
#define READ 1  
#define WRITE 0

#define REG_IODIR_A 0x00  
#define REG_IODIR_B 0x10

#define REG_IOCON_A 0x05  
#define REG_IOCON_B 0x15

#define REG_GPIO_A 0x09  
#define REG_GPIO_B 0x19

#define REG_OLAT_A 0x0A  
#define REG_OLAT_B 0x1A

#define ON__MCP23S17_RESET_ACTIVE_LOW 1 // Pin 6, P0_1  
#define OFF__MCP23S17_RESET_ACTIVE_LOW 0 // Pin 6, P0_1

#define ENABLE_5V_REG 0  
#define TRUE 1  
#define FALSE 0

#define BUTTON_PRESSED 77
#define BUTTON_RELEASED 41

/********************************************************************************
* LOCAL VARIABLES
********************************************************************************

// Phase 2 variables
static __xdata DMA_DESC DMA_Channel[NUM_CHANNEL];
static __xdata uint8 FSMstate;
static __xdata uint8 gotoNextState;
static __xdata uint8 SENSOR_RxAgain = FALSE;
static __xdata uint8 SENSOR_Count = 0x00;
static __xdata uint16 SENSOR_Ser = 0x0000;
static __xdata uint8 SENSOR_Col = 0x00;
static __xdata uint8 SENSOR_Loc = 0x00;
static __xdata uint8 Rx_Buff[SENSOR_LEN + 2];
static __xdata uint16 Rx_Sensor[SENSOR_NUM];

// LEDs
static __xdata uint8 reset_LED_i = 0;
static __xdata uint8 reset_LED_2count = 0;
static __xdata uint8 reset_LED_drivePort = 0;
static __xdata uint8 LED_PERSISTENCE_AFTER_SWITCH_RELEASE = 0;
static __xdata uint8 portA_IO = 0x00;
static __xdata uint8 portB_IO = 0x00;
static __xdata uint8 i_count = 0;
static __xdata uint8 scour_reset = 0;
__xdata uint8 reset_LED_sequence[16] = { 0x01, 0x11, 0x01, 0x11,
                                        0x13, 0x33, 0x13, 0x33,
                                        0x37, 0x77, 0x37, 0x77,
                                        0x7F, 0xFF, 0x7F, 0xFF };
// button
static __xdata uint8 force_switch_enable = 0;
static __xdata uint8 prev_button_state = 0xFF;
static __xdata uint8 PUSH_BUTTON_HELD_DOWN = 0;

// timers
static __xdata uint8 timer3_expired = 77;
__xdata uint8 timer1_sec_LUT[2][10];
__xdata uint8 timer1_ms_LUT[2][10];

iscriminations
LOCAL FUNCTIONS
/

// configuration function prototypes
void setup ( ) ;
void setup_spi ( ) ;
void Radio__config ( ) ;

// drive LED function prototype
void reg_write ( uint8 reg_name , uint8 to_write ) ;
void pwr_on_blink_LEDs ( ) ;

// timer function prototypes
void enable_timer1 ( uint8 seconds ) ;
void enable_timer1_100ms_x ( uint8 seconds ) ;
void disable_timer1 ( ) ;

// flash
void halFlashStartErase ( void ) ;
void halFlashStartWrite ( void ) ;
void writeFlashPortInfo ( uint8 dataPort1 , uint8 dataPort2 ) ;
void readFlashPortInfo ( void ) ;
void eraseFlashPortInfo ( void ) ;

#pragma vector = P1_INT_VECTOR
__interrupt void p1_ISR ( void ) ;
T3CTL = T3CTL_DIV_64 | T3CTL_OVFIM | T3CTL_CLR | T3CTL_MODE_FREERUN;
T3OVFIF = 0; T3IF = 0;
T3IE = 1;
T3CTL |= T3CTL_START;

if ((timer3_expired == TRUE) || (timer3_expired == 77)) {
  timer3_expired = FALSE;

  if (P1IFG & BIT3)
  {
    IEN2 &= ~IEN2_P1IE;    // All port1 interrupt DISABLED
    P1IEN &= ~BIT3;        // P1_3's interrupt is DISABLED
    P1IFG &= ~BIT3;        // Clear status flag for pin

    // if == 1 (falling edge)
    // OFF push-button switch (released)
    // NOTE: will not re-enter here if switch is pressed then released
    // again after a press-the-release is initiated.
    if (((PICTL & PICTL_P1ICON) >> (PICTL_P1ICON >> 1)) & 0x01)
    {
      PICTL = (PICTL & ~PICTL_P1ICON); // All port1 rising-edge input
produce interrupt
      disable_timer1();
    }

    PUSH_BUTTON_HELD_DOWN = 0;

    // keep LEDs on for 4 sec after release
    if (((scour_reset == 0) && (prev_button_state ==
        BUTTON_PRESSED)))
    {
      // if the RESET sequence pattern has not started yet (=0)
      if (reset_LED_i == 0)
      {
        LED_PERSISTENCE_AFTER_SWITCH_RELEASE = TRUE;
        enable_timer1(((uint8)4)); // timer to keep LEDs on for 4
        sec after release
      }
    }

  }

  else
  {

  }
}
{  
    // else - DO NOT PERSIST  
    FSMstate = STATE_CONFIG;  
    gotoNextState = 1;  
    force_switch_enable = TRUE;  
    reg_write(REG_GPIO_A, 0x00);  
    reg_write(REG_GPIO_B, 0x00);  
}  
prev_button_state = BUTTON_RELEASED;

reset_LED_2count = 0;  
reset_LED_i = 0;  
reset_LED_drivePort = 0;

//scour_reset is set in T1 after the 16th LED is lit  
//action: clear memory of all scour devices  
if (scour_reset == 1)  
{
    // reset all variations associated with memory of a received  
    float out unit  
    // set scour_reset = 0; before exiting the ISR, not here.  
    // 'else if' below (i.e., the rising edge if-block) if the user is still holding down the button after all 16 lights go out  
    SENSOR_RxAgain = FALSE;  
    SENSOR_Count = 0x00;  
    SENSOR_Ser = 0x0000;  
    SENSOR_Col = 0x00;  
    SENSOR_Loc = 0x00;  
    reset_LED_2count = 0;  
    reset_LED_i = 0;  
    reset_LED_drivePort = 0;  
    writeFlashPortInfo(0x00, 0x00);

    // reset the memory locations of each sensor  
    for (i_count = 0; i_count < SENSOR_NUM; i_count++)  
    {  
        Rx_Sensor[i_count] = 0x0000;  // across all 16 SENSOR_NUM

        if (i_count < (SENSOR_LEN+2))
        {
            //
        }
    }

    }

    prev_button_state = BUTTON_RELEASED;  
}  

{  
    Rx_Buff[i_count] = 0x00;  
}

FSMstate = STATE_CONFIG;  
gotoNextState = 1;  
} // end if scour_reset == 1

// do this just before exiting the ISR  
// disable radio and DMA  
IEN2 &= ~IEN2_RFIE;  
RFIM = 0x00;  
RFIF = 0x00;  
DMAARM |= DMAARM_ABORT;  
DMAIRQ = 0x00;

} // end if == 1 (falling edge)

// if == 0 (rising edge)  
// ON push-button switch (pressed)  
else if ((~(PICTL & PICTL_P1ICON) >> (PICTL_P1ICON >>1)) & 0x01)
{
    disable_timer1();  
    IEN2 &= ~IEN2_RFIE;  
    RFIM = 0x00;  
    RFIF = 0x00;  
    DMAARM |= DMAARM_ABORT;  
    DMAIRQ = 0x00;

    reset_LED_2count = 0;  
    reset_LED_i = 0;  
    reset_LED_drivePort = 0;

    PICTL |= PICTL_P1ICON;  
    // All port1 falling-edge input produces interrupt

    // if no float-out devices communicated yet, drive with "N"  
    if (((portA_IO == 0x00) && (portB_IO == 0x00))
```c
{  
    reg_write(REG_GPIO_A, 0x2F);
    reg_write(REG_GPIO_B, 0xF4);
}
else
{
    reg_write(REG_GPIO_A, portA_IO);
    reg_write(REG_GPIO_B, portB_IO);
}

PUSH_BUTTON_HELD_DOWN = 1;
prev_button_state = BUTTON_PRESSED;
enable_timer1((uint8)10)); // enable timer for 10 sec
// the 10 sec timer tests for initializing
    // the RESET sequence
P1IEN |= BIT3; // P1_3’s interrupt is enabled
IEN2 |= IEN2_P1IE; // All port1 interrupt enabled

// these statements are outside of the rising/falling edge
// if-block statements
if (scour_reset == TRUE)
{
    scour_reset = FALSE;
    // switch interrupt
    P1IEN |= BIT3; // P1_3’s interrupt is enabled
    IEN2 |= IEN2_P1IE; // All port1 interrupt enabled
}

if (force_switch_enable == TRUE)
{
    force_switch_enable = FALSE;
    P1IEN |= BIT3; // P1_3’s interrupt is enabled
    IEN2 |= IEN2_P1IE; // All port1 interrupt enabled
}
```
if ( P1IFG & BIT3 )

else
{
// This block handles software debouncing of the pushbutton switch
NOP(); // debounce condition occurred. Use NOP for breakpoint

PICTL = (PICTL & ~PICTL_P1ICON); // All port1 rising-edge input produces interrupt
    disable_timer1();

    PUSH_BUTTON_HELD_DOWN = 0;

    // keep LEDs on for 4 sec after release
    if ( (scour_reset == 0) && (prev_button_state == BUTTON_PRESSED) )
    {
      if (reset_LED_i == 0)
      {
        LED_PERSISTENCE_AFTER_SWITCH_RELEASE = TRUE;
        enable_timer1(((uint8)4)); // timer to keep LEDs on for 4 sec after release
      }
    }
    else
    {
      // else - DO NOT PERSIST
      FSMstate = STATE_CONFIG;
      gotoNextState = 1;
      force_switch_enable = TRUE;
      reg_write(REG_GPIO_A, 0x00);
      reg_write(REG_GPIO_B, 0x00);
    }

    prev_button_state = BUTTON_RELEASED;
}

reset_LED_2count = 0;
reset_LED_i = 0;
reset_LED_drivePort = 0;

FSMstate = STATE_CONFIG;
gotoNextState = 1;
P1IFG &= ~BIT3;    // Clear status flag for pin
P1IF &= 0;        // Clear CPU interrupt status flag for P0

#pragma vector = T1_VECTOR
__interrupt void TIMER1_ISR(void)
{
    T1IE &= 0;

    if (LED_PERSISTENCE_AFTER_SWITCH_RELEASE == TRUE)
    {
        disable_timer1();

        T3IE = 0;
        T3OVFIF &= 0;
        T3IF &= 0;
        T3CTL = (T3CTL & ~((T3CTL_START | T3CTL_OVFIM)) | T3CTL_CLR);

        // disable switch interrupts and clear flags just to establish a
        // cleared baseline
        IEN2 &= ~IEN2_P1IE;    // All port1 interrupt DISABLED
        P1IFG &= ~BIT3;        // P1_3’s interrupt is DISABLED
        P1IF &= 0;

        // disable radio and DMA
        IEN2 &= ~IEN2_RFIE;
        RFIM = 0x00;
        RFIF = 0x00;
        DMAARM | = DMAARM_ABORT;
        DMAIRQ = 0x00;

        // end persistence - clear all LEDs
        reg_write(REG_GPIO_A, 0x00);
        reg_write(REG_GPIO_B, 0x00);

        LED_PERSISTENCE_AFTER_SWITCH_RELEASE = FALSE;
        PUSH_BUTTON_HELD_DOWN = FALSE;
// strobe to radio state
FSMstate = STATE_CONFIG;
goToNextState = 1;

// switch interrupt
PIEN |= BIT3;  // PI_3's interrupt is enabled
IEN2 |= IEN2_PIIE;  // All port1 interrupt enabled

// IF THE PUSH-BUTTON IS BEING HELD DOWN
if (PUSH_BUTTON_HELD_DOWN == 1)
{
    disable_timer1();

    T3IE = 0;
    T3OVFIF = 0;
    T3IF = 0;
    T3CTL = (T3CTL & ~(T3CTL_START | T3CTL_OVFIM)) | T3CTL_CLR;

    // clear LED matrix
    if (reset_LED_i == 0)
    {
        reg_write(REG_GPIO_A, 0x00);
        reg_write(REG_GPIO_B, 0x00);
    }

    if (++reset_LED_2count > 2)
    {
        reset_LED_2count = 1;  // reset to 1 and not 0 because you do
                                 // indeed drive a port below after
                                 // resetting reset_LED_2count
        reset_LED_drivePort ^= 1;  // toggle port to drive
    }

    // check LEDs and drive accordingly,
    // check if RESET/CLEAR sequence is being initiated
    // if # of LEDs driven is less than full matrix
    if (reset_LED_i <= 15)
    {
        if (reset_LED_drivePort == 0)
```c
425  {
        reg_write(REG_GPIO_A, reset_LED_sequence[reset_LED_i++]);
        enable_timer1((uint8)1);
    }
    else
    {
        reg_write(REG_GPIO_B, reset_LED_sequence[reset_LED_i++]);
        enable_timer1((uint8)1);
    }
    else // if all LEDs are driven then reset everything
    {
        // this occurs after the 16th LED is lit
        disable_timer1();

        reset_LED_2count = 0;
        reset_LED_i = 0;
        reset_LED_drivePort = 0;

        PUSH_BUTTON_HELD_DOWN = 0;
        reg_write(REG_GPIO_A, 0x00);
        reg_write(REG_GPIO_B, 0x00);
        portA_IO = 0x00;
        portB_IO = 0x00;
        scour_reset = 1;
    } // end if (PUSH_BUTTON_HELD_DOWN == 1)

#pragma vector = T3_VECTOR
__interrupt void t3_isr(void)
{
    T3IE = 0;
    T3OVFIF = 0;
    T3IF = 0;
    T3CTL = (T3CTL & ~(T3CTL_START | T3CTL_OVFIM)) | T3CTL_CLR;

    timer3_expired = TRUE;
}
#pragma vector = RF_VECTOR
__interrupt void RF_IRQ(void)
```


```c
{  S1CON &= ~0x03; // Clear the cpu RF interrupt flag

  if (RFIF & RFIF_IRQ_DONE)
  {
    RFIF &= ~RFIF_IRQ_DONE; // Clear RF Timeout Interrupt Flag
    DMAIRQ &= ~DMAIRQ_DMAIF0; // Clear DMA Channel 0 Interrupt Flag

    if (PKTSTATUS & 0x80)
    {
      // Check if the RFRX Packet CRC was correct
      if (Rx_Buff[0] == BrID_0 && Rx_Buff[1] == BrID_1 && Rx_Buff[2] == BrID_2
        { // Check if the Bridge ID was correct
          // Store the Float-out Sensor Serial Numbers from the received data
          SENSOR_Ser = (0xFF00 & (Rx_Buff[8] << 8)) | (0x00FF & (Rx_Buff[9] << 0));

          // Reset SensorReceivedAgain Semaphore for Redundancy
          SENSOR_RxAgain = FALSE;

          // Loop and Check if Sensor Serial Number has already been Received
          for (int i = 0; i < SENSOR_NUM; i++)
          {
            if (SENSOR_Ser == Rx_Sensor[i])
              { // Sensor has been previously received
                SENSOR_RxAgain = TRUE;
                break;
              }
          }

          if (SENSOR_RxAgain == TRUE)
            { // If Sensor has previously been received, Reset semaphore and Reconfigure Radio/DMA
              SENSOR_RxAgain = FALSE;
              FSMstate = STATE_CONFIG;
            }
          else
            { // If new float-out device is received, update the LED patterns
```
FSMstate = STATE_LED;
}
else
{/ / If Bridge ID does not match – Reconfigure Settings for Radio/DMA

FSMstate = STATE_CONFIG;
}
else
{/ / If RF RX Packet CRC was incorrect – Reconfigure Settings for Radio/DMA

FSMstate = STATE_CONFIG;
} gotoNextState = TRUE; // Continue onto next state
}

int main(void)
{
EA = 1;
setup();
P2_3 = ENABLE_5V_REG; // enable 5V regulator for LED matrix and I/O Expander
enable_timer1((uint8)3); // 3 second delay
while ( !( T1CTL & T1CTL_OVFIF ) );
disable_timer1();
P0_1 = ON__MCP23S17_RESET_ACTIVE_LOW; // reset I/O Expander to put into a known state
enable_timer1(2); // 2 second delay
while ( !( T1CTL & T1CTL_OVFIF ) );
disable_timer1();
P0_1 = OFF__MCP23S17_RESET_ACTIVE_LOW; // bring I/O Expander out of reset
enable_timer1(1); // 1 second delay
setup_spi();
// Initialize LED port registers by retrieving their last known state
// from FLASH memory.
readFlashPortInfo();
if( (read_flash_data[0] == 0xFF) && (read_flash_data[1] == 0xFF))
{
    portA_IO = 0x00;
    portB_IO = 0x00;

    writeFlashPortInfo(portA_IO, portB_IO);
}
else
{
    portA_IO = read_flash_data[0];
    portB_IO = read_flash_data[1];
}

// set-up I/O Expander port registers
reg_write(REG_IOCNA, 0xB0);
reg_write(REG_IODIRA, 0x00);
reg_write(REG_OLATA, 0xFF);
reg_write(REG_GPIOA, 0x00);
reg_write(REG_IOCNB, 0xB0);
reg_write(REG_IODIRB, 0x00);
reg_write(REG_OLATB, 0xFF);
reg_write(REG_GPIOB, 0x00);

// Flash the Start-up LED sequence/pattern to signal that
// the receiver is now ready for operator control.
pwr_on_blink_LEDS();

disable_timer1();

FSMstate = STATE_CONFIG;

P1IFG &= ~BIT3;       // Clear status flag for pin
P1IF = 0;             // Clear CPU interrupt status flag for P1

P1IEN |= BIT3;        // P1_3’s interrupt is enabled
IEN2 |= IEN2_P1IE;    // All port1 interrupt enabled
while (1)
{
    switch (FSMstate)
    {
        case STATE_CONFIG:
        {
            DMA_Channel[0].SRCADDRH = (uint16)(&X_RFD) >> 8;
            DMA_Channel[0].SRCADDRL = (uint16)(&X_RFD);
            DMA_Channel[0].DESTADDRH = ((uint16)&Rx_Buff) >> 8;
            DMA_Channel[0].DESTADDRL = ((uint16)&Rx_Buff);
            DMA_Channel[0].VLEN = DMA_VLEN_FIXED;
            DMA_Channel[0].LENH = 0;
            DMA_Channel[0].LENL = SENSOR_LEN + 2;
            DMA_Channel[0].TRIG = DMA_TRIG_RADIO;
            DMA_Channel[0].WORDEIZE = DMA_WORDSIZE_BYTE;
            DMA_Channel[0].TMODE = DMA_TMODE_SINGLE;
            DMA_Channel[0].SRCINC = DMA_SRCINC_0;
            DMA_Channel[0].DESTINC = DMA_DESTINC_1;
            DMA_Channel[0].IRQMASK = DMA_IRQMASK_DISABLE;
            DMA_Channel[0].M8 = DMA_M8_USE_8_BITS;
            DMA_Channel[0].PRIORITY = DMA_PRI_HIGH;
            DMA0CFGL = (uint16)((DMA_Channel[0]));
            DMA0CFGH = (uint16)((DMA_Channel[0])) >> 8;

            IP1 |= IP1_IPG0; IP0 |= IP0_IPG0; // set IPG3 to highest priority
            IP1 |= IP1_IPG1; IP0 &= ~IP0_IPG1; // set IPG0 to second lowest priority
            Radio__config();

            RFST = RFST_SIDLE; // Strobe radio to IDLE STATE
            while(MARCSTATE != MARC_STATE_IDLE); // wait until Radio enters idle state

            RFST = RFST_SCAL; // Strobe radio to calibrate frequency synthesizer
while (MARCSTATE != MARC_STATE_IDLE); \/
\quad // wait until Radio enters idle state

FSMstate = STATE_RFRX;
gotoNextState = TRUE;
} break;

case STATE_RFRX:
{
    PKTLEN = SENSOR_LEN; \/
\quad // Set RF Packet Lengths

    RFIF = 0x00; \/
\quad // Clear RF IRQ flags
    IEN2 |= IEN2_RFIE; \/
\quad // RF Interrupts Enabled
    RFIM = RFIF_IRQ_DONE; \/
\quad // Interrupt on Tx/Rx Completed
    EA = 1; \/
\quad // Enable General Interrupts

    DMAARM |= DMA_CHANNEL_0;
    NOP(); NOP(); NOP(); NOP(); NOP(); NOP(); NOP(); NOP();

    RFST = RFST_SRX; \/
\quad // Strobe radio into Rx
    while ((MARCSTATE != MARC_STATE_RX)); \/
\quad // Wait for Radio to enter RX
} break;

case STATE_LED:
{
    SENSOR_Ser = (0xFF00 & (Rx_Buff[8] «<8)) | (0x00FF & (Rx_Buff
[9] «<0));
    SENSOR_Loc = (Rx_Buff[10] & 0xF0);
    SENSOR_Col = (Rx_Buff[10] & 0x0F);

    // Record the sensor serial number
    Rx_Sensor[SENSOR_Count++] = SENSOR_Ser;

    // Set the I/O Expander’s portA or portB bit pattern
    // which corresponds to the received sensor location
    // and sensor column
    if (SENSOR_Loc == SCOUR_LOC1)
    {
        if (SENSOR_Col == SCOUR_GREEN)
        {

            // More code here
        }
    }

    // More code here
if ((portA_IO & 0x0E) == 0)
{
    portA_IO = ((portA_IO & 0xF0) | 0x01);
}
else if(SENSOR_Col == SCOUR_YELLOW)
{
    if ((portA_IO & 0x0C) == 0)
    {
        portA_IO = ((portA_IO & 0xF0) | 0x02);
    }
}
else if(SENSOR_Col == SCOUR_ORANGE)
{
    if ((portA_IO & 0x08) == 0)
    {
        portA_IO = ((portA_IO & 0xF0) | 0x04);
    }
    else if(SENSOR_Col == SCOUR_RED)
    {
        portA_IO = ((portA_IO & 0xF0) | 0x08);
    }
}
else if(SENSOR_Loc == SCOUR_LOC2)
{
    if(SENSOR_Col == SCOUR_GREEN)
    {
        if ((portA_IO & 0xE0) == 0)
        {
            portA_IO = ((portA_IO & 0xF0) | 0x10);
        }
    }
    else if(SENSOR_Col == SCOUR_YELLOW)
    {
        if ((portA_IO & 0xC0) == 0)
        {
            portA_IO = ((portA_IO & 0xF0) | 0x20);
        }
    }
    else if(SENSOR_Col == SCOUR_ORANGE)
if ((portA_IO & 0x80) == 0)
  portA_IO = ((portA_IO & 0xF0) | 0x40);
else if(SENSOR_Col == SCOUR_RED)
  portA_IO = ((portA_IO & 0xF0) | 0x80);
else if(SENSOR_Loc == SCOUR_LOC3)
  if(SENSOR_Col == SCOUR_GREEN)
    if ((portB_IO & 0x0E) == 0)
      portB_IO = ((portB_IO & 0xF0) | 0x01);
  else if(SENSOR_Col == SCOUR_YELLOW)
    if ((portB_IO & 0x0C) == 0)
      portB_IO = ((portB_IO & 0xF0) | 0x02);
  else if(SENSOR_Col == SCOUR_ORANGE)
    if ((portB_IO & 0x08) == 0)
      portB_IO = ((portB_IO & 0xF0) | 0x04);
  else if(SENSOR_Col == SCOUR_RED)
    portB_IO = ((portB_IO & 0xF0) | 0x08);
else if(SENSOR_Loc == SCOUR_LOC4)
if (SENSOR_Col == SCOUR_GREEN)
{
    if ((portB_IO & 0xE0) == 0)
    {
        portB_IO = ((portB_IO & 0xF) | 0x10);
    }
}
else if (SENSOR_Col == SCOUR_YELLOW)
{
    if ((portB_IO & 0xC0) == 0)
    {
        portB_IO = ((portB_IO & 0xF) | 0x20);
    }
}
else if (SENSOR_Col == SCOUR_ORANGE)
{
    if ((portB_IO & 0x80) == 0)
    {
        portB_IO = ((portB_IO & 0xF) | 0x40);
    }
}
else if (SENSOR_Col == SCOUR_RED)
{
    portB_IO = ((portB_IO & 0xF) | 0x80);
}

// write the portA and portB bit patterns to FLASH memory
writeFlashPortInfo(portA_IO, portB_IO);

FSMstate = STATE_CONFIG;
gotoNextState = TRUE;
}
break;

default:
{
    FSMstate = STATE_CONFIG;
gotoNextState = TRUE;
}break;

while (gotoNextState != TRUE);
gotoNextState = FALSE;
}
return 0;
}

void setup()
{
    // MASTER CLOCK
    // high speed xtal osc
    // tickspd = f_ref/128 = 203.125kHz
    // clkspd = fref = 26 MHz
    SLEEP &= ~SLEEP_OSC_PD;
    while ( !(SLEEP & SLEEP_XOSC_S) ) ;
    CLKCON = (CLKCON & ~CLKCON_OSC) | TICKSPD_DIV_64 | CLKSPD_DIV_1;
    while (CLKCON & CLKCON_OSC);
    SLEEP |= SLEEP_OSC_PD;

    // 5V Regulator I/O Configuration
    P2SEL &= ~P2SEL_SEL&P2_3; // P2_3 is the enable for 5V regulator
    P2DIR |= (BIT3); // P2_3 is the enable for 5V regulator

    // push-button switch
    P1SEL &= ~BIT3; // P1_3 is switch input
    P1DIR &= ~BIT3; // P1_3 is input
    P1INP &= ~BIT3; // P1_3 is pull-up/down // 0= pull-up/down, 1 = tristate
    P2INP |= P2INP_PDUP1; // All port1 is pull-down
    PICTL = (PICTL & ~PICTL_P1ICON); // All port1 rising-edge input produces interrupt

    // RESET of MCP23S17 (active low) is P0_1
    P0SEL = (P0SEL & ~BIT1); // P0_1 is GPIO
    P0DIR |= BIT1; // P0_1 is output
    P0INP = (P0INP & ~BIT1); // P0_1 is pull-up/down
    P2INP = (P2INP & ~P2INP_PDUP0); // All port0 is pull-up

    // timer1 period = 26MHz/128/128 = 1.587kHz per bit count
    // timer1 bit period = 0.00063015384615384615384615384615385 seconds;
timer1_bit_freq_H = 1587;

uint16 timer1_bit_freq = 0x0319;  // = 1 sec because the counter counts up
then down before overflow is reached

uint8 t1_count = 0;

// populate the 1 sec resolution (from 1 to 10 seconds) look-up table.
for (t1_count = 1; t1_count <= 10; t1_count++)
{
    timer1_sec_LUT[0][t1_count-1] = ((uint8)( (timer1_bit_freq * t1_count) & 0x00FF ));
    timer1_sec_LUT[1][t1_count-1] = ((uint8)( ( (timer1_bit_freq * t1_count) >> 8 ) & 0x00FF ));

    timer1_ms_LUT[0][t1_count-1] = ((uint8)( ( (timer1_bit_freq / 10) * t1_count ) & 0x00FF ));
    timer1_ms_LUT[1][t1_count-1] = ((uint8)( ( (timer1_bit_freq / 10) * t1_count ) >> 8 ) & 0x00FF ));
}

void setup_spi()
{
    //SPI configuration registers
    // assign USART0 to "Alternative 2 location" (Port1 pins 5,4,3,2)
    // assign USART1 to "Alternative 1 location" (Port0 pins 5,4,3,2)
    PERCFG = (PERCFG & ~PERCFG_U1CFG) | PERCFG_U0CFG;

    // port priority in case of assign conflict
    P2DIR = (P2DIR & ~P2DIR_PRIP0) | P2DIR_PRIP0_0;

    // use USART1 on Port 0 with pins 5,4,3,2
    P0SEL = (P0SEL & ~BIT2) | BIT5 | BIT4 | BIT3;

    // P0.2 is the SS/CS line of the SPI
    P0DIR |= BIT2;

    // USART1 mode = SPI, SPI master
U1CSR &= ~(U1CSR_MODE | U1CSR_SLAVE);

// CPOL = negative clock polarity
// CPHA = data centered on first clock edge
// ORDER = MSB first
// SPI baud rate = 4.01171875 MHz
U1BAUD = SPI_BAUD_M;
U1GCR = (U1GCR & ~(U1GCR_BAUD_E | U1GCR_CPOL | U1GCR_CPHA)) | SPI_BAUD_E;
U1GCR |= U1GCR_ORDER;

// function to write a byte to an 8-bit register of the I/O Expander
void reg_write(uint8 reg_name, uint8 to_write)
{
    P0_2 = 0; // SS low

    U1DBUF = ((0x04)<<4) | ((SLAVE_ADDR_EXT)<<1) | WRITE;
    while(!(U1CSR & U1CSR_TX_BYTE)); // Check if byte is transmitted
    U1CSR &= ~U1CSR_TX_BYTE; // Clear transmit byte status
    U1DBUF = reg_name;
    while(!(U1CSR & U1CSR_TX_BYTE)); // Check if byte is transmitted
    U1CSR &= ~U1CSR_TX_BYTE; // Clear transmit byte status
    U1DBUF = to_write;
    while(!(U1CSR & U1CSR_TX_BYTE)); // Check if byte is transmitted
    U1CSR &= ~U1CSR_TX_BYTE; // Clear transmit byte status

    P0_2 = 1; // SS low
}

// radio configuration function
void Radio__config()
{
    SYNC1 = 0xD3; // sync word, high byte
    SYNC0 = 0x91; // sync word, low byte
    PKTLEN = 0x0B; // packet length
    PKTCTRL1 = 0x04; // packet automation control
    PKTCTRL0 = 0x44; // packet automation control --0x05
    ADDR = 0x00; // device address
    CHANNR = 0x00; // channel number
FSCTRL1 = 0x06; // frequency synthesizer control
FSCTRL0 = 0x00; // frequency synthesizer control
FREQ2 = 0x23; // frequency control word, high byte
FREQ1 = 0x31; // frequency control word, middle byte
FREQ0 = 0x3B; // frequency control word, low byte

MDMCFG4 = 0x2D; // modem configuration
MDMCFG3 = 0x3B; // modem configuration
MDMCFG2 = 0x03; // modem configuration
MDMCFG1 = 0x22; // modem configuration
MDMCFG0 = 0xF8; // modem configuration

DEVIATN = 0x51; // modem deviation setting
MCSM2 = 0x07; // main radio control state machine configuration
MCSM1 = 0x30; // main radio control state machine configuration
MCSM0 = 0x18; // main radio control state machine configuration

FOCCFG = 0x17; // frequency offset compensation configuration
BSCFG = 0x6C; // bit synchronization configuration
AGCCTRL2 = 0x03; // agc control
AGCCTRL1 = 0x40; // agc control
AGCCTRL0 = 0x91; // agc control
FREND1 = 0x56; // front end rx configuration
FREND0 = 0x10; // front end tx configuration
FSCAL3 = 0xEA; // frequency synthesizer calibration
FSCAL2 = 0x2A; // frequency synthesizer calibration
FSCAL1 = 0x00; // frequency synthesizer calibration
FSCAL0 = 0x1F; // frequency synthesizer calibration
TEST2 = 0x88; // various test settings
TEST1 = 0x31; // various test settings
TEST0 = 0x09; // various test settings
PA_TABLE0 = 0x8E; // pa power setting 0

// enable timer1 for X seconds, where X is the function's argument (unsigned 8-bit int)
inline void enable_timer1(uint8 seconds)
{
    T1IE = 0;
    T1CTL &= ~T1CTL_OVFIF;
    T1CTL &= T1CTL_MODE_SUSPEND;
}
// assign look-up table's value to timer1 registers
T1CCOL = timer1_sec_LUT[0][(seconds - 1)];
T1CCOH = timer1_sec_LUT[1][(seconds - 1)];
T1CTL = (T1CTL & ~(T1CTL_DIV)) | T1CTL_MODE_UPDOWN | T1CTL_DIV_128 | T1CTL_MODE;

OVFM = 1;
T1IE = 1;
}

// enable timer1 for X *100ms, where X is the function's argument (unsigned 8-bit int)
inline void enable_timer1_100ms_x(uint8 seconds)
{
    T1IE = 0;
    T1CTL &= ~T1CTL_OVFIF;
    T1CTL &= T1CTL_MODE_SUSPEND;

    // assign look-up table's value to timer1 registers
    T1CCOL = timer1_ms_LUT[0][(seconds - 1)];
    T1CCOH = timer1_ms_LUT[1][(seconds - 1)];
    T1CTL = (T1CTL & ~(T1CTL_DIV)) | T1CTL_MODE_UPDOWN | T1CTL_DIV_128 | T1CTL_MODE;

    OVFM = 1;
    T1IE = 1;
}

// timer1 disable function
inline void disable_timer1()
{
    T1CNTL = 0x00;
    T1IE = 0;
    T1CTL &= ~T1CTL_OVFIF;
    T1CTL &= T1CTL_MODE_SUSPEND;
    OVFM = 0;
}
void pwr_on_blink_LEDs()
{
    uint8 num_blinks = 0;
    for (num_blinks = 0; num_blinks < 2; num_blinks++)
    {
        reg_write(REG_GPIO_A, 0x5A);
        reg_write(REG_GPIO_B, 0x5A);
        enable_timer1_100ms_x(2);
        while (!(T1CTL & T1CTL_OVFIF));
        disable_timer1();

        reg_write(REG_GPIO_A, 0x00);
        reg_write(REG_GPIO_B, 0x00);
        enable_timer1_100ms_x(2);
        while (!(T1CTL & T1CTL_OVFIF));
        disable_timer1();

        reg_write(REG_GPIO_A, 0xA5);
        reg_write(REG_GPIO_B, 0xA5);
        enable_timer1_100ms_x(2);
        while (!(T1CTL & T1CTL_OVFIF));
        disable_timer1();

        reg_write(REG_GPIO_A, 0x00);
        reg_write(REG_GPIO_B, 0x00);
        enable_timer1_100ms_x(2);
        while (!(T1CTL & T1CTL_OVFIF));
        disable_timer1();
    }
}

void writeFlashPortInfo(uint8 dataPort1, uint8 dataPort2)
{
    write_flash_data[0] = dataPort1;
    write_flash_data[1] = dataPort2;

    DMA_Channel[1].SRCHADDH = ((uint16)write_flash_data >> 8) & 0x00FF;
    DMA_Channel[1].SRCADDDR = (uint16)write_flash_data & 0x00FF;
DMA_Channel[1].DESTADDRH = ((uint16)&X_FWDATA >> 8) & 0x00FF;
DMA_Channel[1].DESTADDRL = (uint16)&X_FWDATA & 0x00FF;
DMA_Channel[1].VLEN = DMA_VLEN_USE_LEN;
DMA_Channel[1].LENH = (FLASH_DATA_LEN >> 8) & 0x00FF;
DMA_Channel[1].LENL = FLASH_DATA_LEN & 0x00FF;
DMA_Channel[1].WORDSIZEx = DMA_WORDSIZE_BYTE;
DMA_Channel[1].TMODE = DMA_TMODE_SINGLE;
DMA_Channel[1].TRIG = DMA_TRIG_FLASH;
DMA_Channel[1].SRCINC = DMA_SRCINC_1;
DMA_Channel[1].DESTINC = DMA_DESTINC_0;
DMA_Channel[1].IRQMASK = DMA_IRQMASK_DISABLE;
DMA_Channel[1].MS = DMA_MS_USE_8_BITS;
DMA_Channel[1].PRIORITY = DMA_PRI_HIGH;

DMA1CFGH = ((uint16)&DMA_Channel[1] >> 8) & 0x00FF;
DMA1CFGL = (uint16)&DMA_Channel[1] & 0x00FF;

eraseFlashPortInfo();

// Waiting for the flash controller to be ready
while (FCTL & FCTL_BUSY);

// Set Flash Write Timing (FWT) to 0x22 = 26 MHz
FWT = 0x22;
FADDRH = (int)flashDataAddr >> 9;
FADDRL = ((int)flashDataAddr >> 1) & ~0xFF00;

// Arm the DMA channel, so that a DMA trigger will initiate DMA writing
DMAARM = DMA_ARM1;

// Enable flash write. Generates a DMA trigger. Must be aligned on a 2-byte boundary
halFlashStartWrite();

// Wait for DMA transfer to complete
while (!(DMAIRQ & DMAIRQ_DMAIF1));

// Wait until flash controller not busy
while (FCTL & (FCTL_BUSY | FCTL_SWBSY));

// By now, the transfer is completed, so the transfer count is reached.
// The DMA channel 1 interrupt flag is then set, so we clear it here.
DMAIRQ &= ~DMAIRQ_DMAIF1;
}

// read portA and portB data residing in FLASH memory
void readFlashPortInfo(void)
{
    // Read from flash to check whether the write was successful.
    uint8 i;

    for (i = 0; i < FLASH_DATA_LEN; i++)
    {
        read_flash_data[i] = flashDataAddr[i];
    }

    // clear the page where portA and portB reside in FLASH memory
void eraseFlashPortInfo(void)
{
    /* Waiting for the flash controller to be ready */
    while (FCTL & FCTL_BUSY);

    // Set Flash Write Timing (FWT) to 0x22 = 26 MHz
    FWT = 0x22;
    FADDRH = (int)flashDataAddr >> 9;
    FADDRL = ((int)flashDataAddr >> 1) & ~0xFF00;

    // Erase the page that will be written to. */
    halFlashStartErase();

    // Wait for the erase operation to complete. */
    while (FCTL & FCTL_BUSY);
}
/*************** INCLUDES *******************/
#include <hal_defs.h>
#include "hal_types.h"
#include <hal_cc8051.h>
#include "ioCCxx10_bitdef.h"
#include "dma.h"
#include <ioCC1110.h>

/******* BRIDGE INFO *******
#define BrID_0 0x8B
#define BrID_1 0x54
#define BrID_2 0x6F
#define BrID_3 0x6D
#define BrID_4 0xD2
#define BrID_5 0x02
#define BrID_6 0x00
#define BrID_7 0x00

/*************** SCOUR LOCATIONs ***************
#define SCOUR_LOC1 0xA0
#define SCOUR_LOC2 0xB0
#define SCOUR_LOC3 0xC0
#define SCOUR_LOC4 0xD0

/*************** SCOUR COLORS ***************
#define SCOUR_GREEN 0x01
#define SCOUR_YELLOW 0x02
#define SCOUR_ORANGE 0x03
#define SCOUR_RED 0x04

/******* LED Flashing Period *******
#define TIMER1_H 0x7B  // 5 Seconds
#define TIMER1_L 0xFA  // 5 Seconds
#define TIMER1_COUNT 0x8700 // 24 Hours

/**************************** ARRAY DECLARATION ******************************/

#define SENSOR_LEN 11
#define SENSOR_NUM 32

//static __xdata uint8 Rx_Buff[SENSOR_LEN + 2];
//static __xdata uint16 Rx_Sensor[SENSOR_NUM];

/**************************** STATE DECLARATION ******************************/

#define STATE_CONFIG 0
#define STATE_RFRX 1
#define STATE_LED 2
#define STATE_DEBUG 3

#define TRUE 1
#define FALSE 0

#define NUM_CHANNEL 2

/****************************-----------------------------------------------*/

code/scour_receiver_h_phase3.h

/***************************************************************************/

Filename : flash_asm.s51

Description: This code implements the low level setup and triggering of FLASH Erase and Write. Used by the [flash_dma.c] and [flash_cpu].

Note: This file was provided by Texas Instruments.

***************************************************************************/
includes

#include "ioCC2510.h"

 constants

 local variables

 local functions

 module flash_asm.s51 ; file name

/*
 * halFlashStartErase()
 */

 RSEG RCODE(1) ; 2-byte alignment
 PUBLIC halFlashStartErase
 FUNCTION halFlashStartErase, 0203H

 halFlashStartErase:
 ORL FCTL, #0x01; ; FCTL | = FCTL_ERASE
 NOP; ; Required, see datasheet.
 RET;
/* halFlashStartWrite() */

RSEG RCODE(1);  // 2-byte alignment

PUBLIC halFlashStartWrite

FUNCTION halFlashStartWrite, 0203H

halFlashStartWrite:

ORL FCTL, #0x02;  // FCTL |= FCTL_WRITE

RET;

END;

/*****************************************************************************/

Copyright 2008 Texas Instruments Incorporated. All rights reserved.

IMPORTANT: Your use of this Software is limited to those specific rights
granted under the terms of a software license agreement between the user
who downloaded the software, his/her employer (which must be your employer)
and Texas Instruments Incorporated (the "License"). You may not use this
Software unless you agree to abide by the terms of the License. The License
limits your use, and you acknowledge, that the Software may not be modified,
copied or distributed unless embedded on a Texas Instruments microcontroller
or used solely and exclusively in conjunction with a Texas Instruments radio
frequency transceiver, which is integrated into your product. Other than
for
the foregoing purpose, you may not use, reproduce, copy, prepare derivative
works of, modify, distribute, perform, display or sell this Software and/or
its documentation for any purpose.

YOU FURTHER ACKNOWLEDGE AND AGREE THAT THE SOFTWARE AND DOCUMENTATION ARE
PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED,
INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY, TITLE,
NON-INFRINGEMENT AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL
TEXAS INSTRUMENTS OR ITS LICENSORS BE LIABLE OR OBLIGATED UNDER CONTRACT,
NEGLIGENCE, STRICT LIABILITY, CONTRIBUTION, BREACH OF WARRANTY, OR OTHER
LEGAL EQUITABLE THEORY ANY DIRECT OR INDIRECT DAMAGES OR EXPENSES
INCLUDING BUT NOT LIMITED TO ANY INCIDENTAL, SPECIAL, INDIRECT, PUNITIVE
OR CONSEQUENTIAL DAMAGES, LOST PROFITS OR LOST DATA, COST OF PROCUREMENT
OF SUBSTITUTE GOODS, TECHNOLOGY, SERVICES, OR ANY CLAIMS BY THIRD PARTIES
(INCLUDING BUT NOT LIMITED TO ANY DEFENSE THEREOF), OR OTHER SIMILAR COSTS.

Should you have any questions regarding your right to use this Software,
contact Texas Instruments Incorporated at www.TI.com.

***********************************************************************************

code/flash_asm.s51
Appendix: Float-out Device Embedded Code

The following refers to Section 4.1:

To uniquely identify each float-out device (i.e., unique identification number, scour color depth indicator, and scour hole number), two “#define” statements in the header file (scour_sensor_h_phase3.h) must be modified. In the header file, notice the definitions “#define SENSORSER_8” and “#define SENSORSER_9” on line 19–20, which correspond to the unique identification number of a float out device. In this example, the float-out device was programmed with 0x0103 (i.e., = decimal 259), where SENSORSER_8 is the most significant byte. Next, notice the definitions, “#define ORANGE” and “#define LOC1” on line 22 and line 25, respectively. These definitions define the scour color depth indicator, in this case “ORANGE”, and the scour hole number, in this case “LOC1”. Therefore, the float-out device programmed in this example will have the unique identification number 259, it will be in scour hole number 1, and it will be at an orange scour color depth. Refer to row 3 column 1 in Figure 8 so see the how this programming reflects the physical scour location and light indicator LED in the receiver unit LED matrix.

The bridge identification number (i.e., decimal 3102802400395; 0x000002D26D6F548B) is stored in BRIDGEID_7 through BRIDGEID_0, where BRIDGE_7 is the most significant byte. The bridge identification number is the same across all float-out devices in this Phase 3 installation.
#include <scour_Sensor.h>

#define DEBUG_LED 1

/* #define DEBUG_SHUTDOWN 1 */

/* LOCAL VARIABLES */

static __xdata uint8 FSMstate;
static __xdata uint8 gotoNextState;

static __xdata DMA_DESC DMA_Channel[NUM_CHANNEL];

static __xdata uint16 T1_Val = 0x0000;
static __xdata uint16 RandNum = 0x0000;
static __xdata uint8 RandSeed = TRUE;
static __xdata uint16 shutdownCount = 0x0000;

/* LOCAL FUNCTIONS */

void Radio__config(void);

int main(void)
{
    SLEEP &= ~SLEEP_OSC_PD;
    while( !(SLEEP & SLEEP_XOSC_S) );
    CLKCON = (CLKCON & ~CLKCON_OSC) | CLKCON_OSC32 | TICKSPD_DIV_32 | CLKSPD_DIV_1;
    while (CLKCON & CLKCON_OSC);
}
SLEEP |= SLEEP_OSC_PD;

// Configure P0_0/1/2 GPIO Output – LED and Shutdown Control
P0SEL &= ~(BIT0 | BIT1 | BIT2);  // Configure P0_0/1/2 as GPIO
P0DIR |= (BIT0 | BIT1 | BIT2);  // Configure P0_0/1/2 as Output
P0_2 = 0;  // Configure LEDs and Shutdown Pins

#ifdef DEBUG_LED
    P0_0 = 0; P0_1 = 1;
#else
    P0_0 = 0; P0_1 = 0;
#endif

// Configure P2_1 GPIO Output – CC1190 HGM(1)/LGM(0) Select
P2SEL &= ~(BIT1);  // Configure P2_1 as GPIO
P2DIR |= (BIT1);  // Configure P2_1 as Output
P2_1 = 0;  // Set CC1190 to LGM

// Configure P2_5 GPIO Output – CC1190 HGM(1)/LGM(0) Select
P1SEL &= ~(BIT5);  // Configure P2_5 as GPIO
P1DIR |= (BIT5);  // Configure P1_5 as Output
P1_5 = 1;  // Set CC1190 to LGM

RandSeed = TRUE;
FSMstate = STATE_CONFIG;

while (1)
{
    switch (FSMstate)
    {
    case STATE_CONFIG:
        { 
            IP1 |= IP1_IPG0; IP0 |= IP0_IPG0;  // set IPG0(RF) to 1st priority
            IP1 |= IP1_IPG1; IP0 &= ~IP0_IPG1;  // set IPG1(T1) to 2nd priority
            IP1 &= ~IP1_IPG1; IP0 |= IP0_IPG1;  // set IPG3(T3) to 3rd priority
            DMA_Channel[1].SRCADDRH = (uint16)(&bridge_ID) >> 8;
            DMA_Channel[1].SRCADDRL = (uint16)(&bridge_ID);
        }
    }
}
DMA_Channel[1].DESTRADDRH = ((uint16)&X_RFD) >> 8;
DMA_Channel[1].DESTRADDR = ((uint16)&X_RFD);
DMA_Channel[1].VLEN = DMA_VLEN_FIXED;
DMA_Channel[1].LENH = 0;
DMA_Channel[1].LENL = BUFF_LEN;
DMA_Channel[1].TRIG = DMA_TRIG_RADIO;
DMA_Channel[1].WORDSIZE = DMA_WORDSIZE_BYTE;
DMA_Channel[1].TMODE = DMA_TMODE_SINGLE;
DMA_Channel[1].SRCINC = DMA_SRCINC_1;
DMA_Channel[1].DESTINC = DMA_DESTINC_0;
DMA_Channel[1].IRQMASK = DMA_IRQMASK_DISABLE;
DMA_Channel[1].BS = DMA_BS_USE_8_BITS;
DMA_Channel[1].PRIORITY = DMA_PRI_HIGH;
DMAICFGH = (uint16)&(DMA_Channel[1]));
DMAICFGH = (uint16)&(DMA_Channel[1])) >> 8;

Radio_config();

shutdownCount = 0x0000;
T3CTL = T3CTL_DIV_128 | T3CTL_START | T3CTL_OVFIM | T3CTL_CLR | T3CTL_MODE_FREERUN;
T3IE = 1; EA = 1;

FSMstate = STATE_WAIT;
gotoNextState = TRUE;
} break;

case STATE_RFTX:
{
   //RFST = RFST_SCAL; //MAR
   // Strobe radio to calibrate frequency synthesizer
   //while(MARCSTATE != MARC_STATE_IDLE); //MAR // wait
   until Radio enters idle state

   DMAIE = 0; DMAIF = 0; // Disable DMA Interrupt and Clear DMA Flag

   PKTLEN = BUFF_LEN; // Set Radio PKTLEN to size of bridge_ID buffer
DMA_ARM |= DMA_CHANNEL_1; // Arm DMA CH1 for RF Transmission

NOP();NOP();NOP();NOP();NOP();NOP();NOP();NOP();NOP();NOP();

RFIF = 0x00; // Clear RF IRQ flags
IEN2 |= IEN2_RFIE; // RF Interrupts Enabled
RFIM = RFIF_IRQ_DONE; // Interrupt on Tx Completed
EA = 1; // Enable General Interrupts

RFST = RFST_SCAL; // Strobe radio to calibrate frequency synthesizer

while (MARCSTATE != MARC_STATE_IDLE); // wait until Radio enters idle state

#ifdef DEBUG_LED
    P0_2 ^= 1;
#endif

RFST = RFST_STX; // Strobe radio into Tx
while ((MARCSTATE != MARC_STATE_TX)); // Wait for Radio to enter RX

} break;

case STATE_WAIT:
{
    // Clear Timer 1 Interrupt Channels 0, 1 and 2
    T1CTL = (T1CTL & ~(T1CTL_CH0IF | T1CTL_CH1IF | T1CTL_CH2IF));

    T1CCCTL0 |= T1CCCTL0_IM | T1CCCTL0_MODE; // Enable Interrupt on Channel 0 & Compare Mode
    T1CCCTL1 &= ~T1CCCTL1_IM; // Disable Interrupt on Channel 1
    T1CCCTL2 &= ~T1CCCTL2_IM; // Disable Interrupt on Channel 1

    T1CTL &= T1CTL_MODE_SUSPEND; // Suspend Operation for Timer 1
    T1CNTL = 0x0000; // Clear the count for Timer 1

    OVFIM = 0; // Disable overflow interrupt
T1IE = 1; EA = 1; // Enable Timer 1 and Global

Interrupts

if (RandSeed == TRUE)
{
    RNDL = SENSORSER_8;
    RNDL = SENSORSER_9;

    RandSeed = FALSE;
}

ADCCON1 |= ADCCON1_RCTRL_LFSR13;
while (ADCCON1 & 0x04);

RandNum = ((RNDH & 0x00FF)<<8) | (RNDL & 0x00FF);

if (RandNum <= T1_DELAY_2T)
{
    T1_Val = 0x0ACD; // Delay Time = 4.2 ms + RF_Cal_Time - 5 ms
}
else if (RandNum > T1_DELAY_2T && RandNum <= T1_DELAY_4T)
{
    T1_Val = 0x1AAB; // Delay Time = 9.2 ms + RF_Cal_Time - 10 ms
}
else if (RandNum > T1_DELAY_4T && RandNum <= T1_DELAY_8T)
{
    T1_Val = 0x3A68; // Delay Time = 19.2 ms + RF_Cal_Time - 20 ms
}
else if (RandNum > T1_DELAY_8T)
{
    T1_Val = 0x79E2; // Delay Time = 39.2 ms + RF_Cal_Time - 40 ms
}
else
{
    T1_Val = 0x0ACD; // Delay Time = 4.2 ms + RF_Cal_Time - 5 ms
}
// Set Timer 1 Compare Registers based on T1_Val
T1CCOH = ((T1_Val >> 8) & 0xFF);
T1CCOL = (T1_Val & 0xFF);

OVFIM = 0; // Disable overflow interrupt
T1IE = 1; EA = 1; // Enable Timer 1 and Global

// Start Timer 1 in Modulo and Tick/1
T1CTL |= T1CTL_MODE_MODULEO | T1CTL_DIV_1;
}
break;

case STATE_SHUTDOWN:
{
    T1CCTL0 &= ~T1CCTL0_IM; // Disable Interrupt for Timer 1 Channel 0
    OVFIM = 0; T1IE = 0; EA = 0; // Disable Overflow, Timer 1 and Global Interrupts
    T1CTL &= T1CTL_MODE_SUSPEND; // Suspend Operation for Timer 1
    T1CNTL = 0x0000; // Clear the count for Timer 1

    // Suspend Timer 3 & Clear Timer 3 Count & Disable Interrupts
    T3CTL = (T3CTL & ~(T3CTL_START | T3CTL_OVFIM)) | T3CTL_CLR;
    T3IE = 0;

#if defined DEBUG_LED
    P0_1 ^= 1; // Toggle LED for Shutdown Timer End
#endif

#if defined DEBUG_SHUTDOWN
    P0_0 = 1; // Drive BJT Current through Reset Coil of Relay
#endif

    FSMstate = STATE_CONFIG;
gotoNextState = TRUE;
}
break;

default:

106
FSMstate = STATE_CONFIG;
gotoNextState = TRUE;
} break;
}

while (gotoNextState != TRUE);
gotoNextState = FALSE;

void Radio__config(void)
{
    /* RF settings CC1110 + CC1190:
    *  – CC1110 Power: 0 dBm
    *  – Output Power: 16.5 dBm
    *  – M3M/LGM Select: LGM
    *  – Data Rate: 250 kbps
    *  – Modulation: 2-FSK
    *  – Deviation: 152 kHz
    *  – MCSM0 = 0x08 – Disable Auto Calibration before Transmit
    */
    SYNC1 = 0xD3; // sync word, high byte
    SYNC0 = 0x91; // sync word, low byte
    PKTLEN = 0x0B; // packet length
    PKTCTRL1 = 0x04; // packet automation control
    PKTCTRL0 = 0x44; // packet automation control
    ADDR = 0x00; // device address
    CHANNR = 0x00; // channel number
    FSCTRL1 = 0x06; // FSCTRL1 = 0x0C; // frequency synthesizer control
    FSCTRL0 = 0x00; // frequency synthesizer control
    FREQ2 = 0x23; // frequency control word, high byte
    FREQ1 = 0x31; // frequency control word, middle byte
    FREQ0 = 0x3B; // frequency control word, low byte
    MDMCFG4 = 0x0D; // MDMCFG4 = 0x2D; // modem configuration
    MDMCFG3 = 0x3B; // modem configuration
    MDMCFG2 = 0x03; // modem configuration
    MDMCFG1 = 0x22; // modem configuration
    MDMCFG0 = 0xF8; // modem configuration
DEVIATN = 0x51; // modem deviation setting
MCSM2 = 0x07; // main radio control state machine configuration
MCSM1 = 0x30; // main radio control state machine configuration
MCSM0 = 0x10; // main radio control state machine configuration
FOCCFG = 0x17; // frequency offset compensation configuration
BSCFG = 0x6C; // bit synchronization configuration
AGCCTRL2 = 0x03; // AGC control
AGCCTRL1 = 0x40; // AGC control
AGCCTRL0 = 0x91; // AGC control
FREND1 = 0x56; // front end rx configuration
FREND0 = 0x10; // front end tx configuration
FSCAL3 = 0xEA; // frequency synthesizer calibration
FSCAL2 = 0x2A; // frequency synthesizer calibration
FSCAL1 = 0x00; // frequency synthesizer calibration
FSCAL0 = 0x1F; // frequency synthesizer calibration

TEST2 = 0x88; // various test settings
TEST1 = 0x31; // various test settings
TEST0 = 0x09; // various test settings

// this is 10dBm output (19.77 dBm measured at CC1190 output) /\[\text{PA_TABLE0 = 0xc0}; /\[\text{PA_TABLE0 = 0x8E}; /\[\text{PA_TABLE0 = 0xC7}; /\[\text{PA_TABLE0 = 0x8E}; // power out = +7 dBm for device #16 (Red Loc 4 Ser. 1028) to compensate for power RX'd at -33 dBm at Spec Analyzer
PA_TABLE0 = 0x83; // power out = +5 dBm for device #3 (Orange Loc 1 Ser.)

// IOCFG2 = 0x00; // radio test signal configuration (p1_7)
// IOCFG1 = 0x00; // radio test signal configuration (p1_6)
// IOCFG0 = 0x06; // radio test signal configuration (p1_5)

#pragma vector=RF_VECTOR
__interrupt void RF_IRQHandler()
{
    SICON &= ~0x03; // Clear the cpu RF interrupt flag

    if (RFIF & RFIF_IRQ_DONE)
    {
        RFIF &= ~RFIF_IRQ_DONE; // Clear RF Timeout Interrupt Flag
        DMAIRQ &= ~DMAIRQ_DMAIF1; // Clear DMA Channel 1 Interrupt Flag

        #ifdef DEBUG_LED
        P0_2 ^= 1;
        #endif

        // FSMstate = STATE_RFTX; //MAR
        FSMstate = STATE_WAIT;
        gotoNextState = TRUE;
    }
}

#pragma vector = T1_VECTOR
__interrupt void TIMER1_IRQHandler()
{
    if (T1CTL & T1CTL_CH0IF)
    {
        // Clear Timer 1 Channel 0 interrupt flag
        T1CTL = (~T1CTL_CH0IF & 0xF0) | (T1CTL & 0x0F);

        // NOTE: VALUE FOR T1CTL_MODE_SUSPEND HAS BEEN CHANGED TO 0xFC. TO SUSPEND THE TIMER WITH &= OPERATION
        T1CTL &= T1CTL_MODE_SUSPEND; // Suspend Operation for Timer 1
        T1CNTL = 0x0000; // Clear the count for Timer 1
        T1CC1L0 &= ~T1CC1L0_IM; T1IE = 0; // Disable Interrupts for Timer 1

        FSMstate = STATE_RFTX;
        gotoNextState = TRUE;
    }
}
```c
__interrupt void TIMER3_ISR(void) {
    T3IF = 0; T3OVFIF = 0; // Clears the CPU and Overflow interrupt flags
    if (shutdownCount++ > T3_SHUTDOWN) {
        // FSMstate = STATE_RFTX; //MAR
        FSMstate = STATE_SHUTDOWN;
        gotoNextState = TRUE;
    }
}
```

```plaintext
code/scour_sensor_c_phase3.c

/******************************************************************************
*********************** Sensor Unit Configuration ***********************
******************************************************************************

/**********************************************************
******************** SENSOR INFO *******************************/
```
# define ORANGE
// # define LOC2
// # define RED
#define LOC1
/**************************
** Birdge Info *************/

/************************** Birdge Info *************/
#define BRIDGEID_0 0xF0
#define BRIDGEID_1 0xE1
#define BRIDGEID_2 0xD2
#define BRIDGEID_3 0xC3
#define BRIDGEID_4 0xB4
#define BRIDGEID_5 0xA5
#define BRIDGEID_6 0x96
#define BRIDGEID_7 0x87

/************************** Birdge Info *************/
#define BRIDGEID_0 0x8B
#define BRIDGEID_1 0x54
#define BRIDGEID_2 0x6F
#define BRIDGEID_3 0x6D
#define BRIDGEID_4 0xD2
#define BRIDGEID_5 0x02
#define BRIDGEID_6 0x00
#define BRIDGEID_7 0x00

#if defined GREEN
 #define SENSORCOL 0x01
#endif
#if defined YELLOW
 #define SENSORCOL 0x02
#endif
#if defined ORANGE
 #define SENSORCOL 0x03
#endif
#if defined AMBER
 #define SENSORCOL 0x05
#endif
#if defined RED
# define SENSORCOL 0x04
#endif

#ifdef LOC1
    #define SENSORLOC 0xA0
#endif

#ifdef LOC2
    #define SENSORLOC 0xB0
#endif

#ifdef LOC3
    #define SENSORLOC 0xC0
#endif

#ifdef LOC4
    #define SENSORLOC 0xD0
#endif

/********************
 TIMER CONFIG
************************
#define T3_SHUTDOWN 498

#ifdef GREEN
    #define T1_DELAY_2T 0x3FFF
    #define T1_DELAY_4T 0x7FFF
    #define T1_DELAY_8T 0xBFFF
#endif

#ifdef YELLOW
    #define T1_DELAY_2T 0x5555
    #define T1_DELAY_4T 0x9555
    #define T1_DELAY_8T 0xD555
#endif

#ifdef ORANGE
    #define T1_DELAY_2T 0x5FFF
    #define T1_DELAY_4T 0xBFFF
    #define T1_DELAY_8T 0xDFFF
#endif

#ifdef AMBER
# define T1_DELAY_2T 0x5FFF
# define T1_DELAY_4T 0xBFFF
# define T1_DELAY_8T 0xDFFF
#endif

#ifdef RED
# define T1_DELAY_2T 0x7FFF
# define T1_DELAY_4T 0xFFFF
# define T1_DELAY_8T 0xFFFF
#endif

/******************** ARRAY DECLARATION **********************/
#define BUFF_LEN 0x0B
#define SENSORINFO (SENSORLOC | SENSORCOL)

static __xdata uint8 bridge_ID[BUFF_LEN] = {BRIDGEID_0, BRIDGEID_1, BRIDGEID_2, BRIDGEID_3, BRIDGEID_4, BRIDGEID_5, BRIDGEID_6, BRIDGEID_7, SENSORER_8, SENSORER_9, SENSORINFO};

/******************** STATE DECLARATION **********************/
#define STATE_CONFIG 0
#define STATE_RFTX 1
#define STATE_WAIT 2
#define STATE_SHUTDOWN 3
#define TRUE 1
#define FALSE 0
#define NUM_CHANNEL 2

/********************/