

**Firmware Release 2.8**

for the Afridev2 PCB 0001-24-00091-01 hardware Rev 2.0

**This firmware version is ONLY for the model explicitly mentioned above. Do not try to update older Afridev2 units using this firmware.**

**Contents**

[Important Notes 2](#_Toc536025103)

[Firmware Build Directions 2](#_Toc536025104)

[Programming Methods 3](#_Toc536025105)

[New and modified Functions 4](#_Toc536025106)

[Fixed Issues 5](#_Toc536025107)

[Known Issues 6](#_Toc536025108)

.

# Important Notes

This package contains the firmware packages for the release. There are three software deliverable items:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Deliverable File | Usage | Programming Method | Contains: | |
| Boot | App |
| 1. **Factory\_App\_Boot\_MSP430.txt** | Factory Test. No Modem connection | MSP430\_Flasher | Yes | Yes |
| 1. **AfridevV2\_App\_Boot.MSP430.txt** | Production Firmware | MSP430\_Flasher | Yes | Yes |
| 1. **AfridevV2\_MSP430\_msg.txt** | Production Firmware | OTA Firmware | No | Yes |
| 1. **AFRIDEV-V2 REMOTE SENSOR**   **MESSAGE SPECIFICATION**  **Version 0.5** | Supported Protocol Version | n/a | n/a | n/a |

# Firmware Build Directions

The firmware was built using the Code Composer Studio Environment Version: **6.1.3.00034.**

The compiler version used is **TI v16.9.10.LTS**

There are 3 projects pertaining to the release: AfridevV2\_MSP430, AfridevV2\_MSP430\_Boot, AfridevV2\_MSP430\_Manuf. All three should be built with the “Debug” Active Configuration. There is no benefit of using the Release Configuration, it is the same.

After running all three builds error and warning free, a Command Prompt Window is opened in Administrator Mode. The current directory is set to the “AfridevV2ImageBuilder” folder within the project’s main folder.

The batch file “run.bat” is executed. If all three builds were run error free, then the tool will succeed and generate the release’s deliverable files in this folder.

# Programming Methods

### MSP430Flasher

The TI MSP 430 Flasher Tool (ver. 1.3.18) can be used to load the flash image of the Firmware. This is used in conjunction with the MSP-FET tool and the MSP-FET-432ADPTR Board. A special harness cable is needed to connect between the MSP FET board and the Afridev2 board.

|  |  |
| --- | --- |
| JTAG Connector | AfriDev Connector |
| J3-P1 (VCC SENSE) | J1-P1 (3.3V) |
| J3-P13 (TDO/SW0) | J1-P2 (nRST) |
| J3-P9 (TCK/SWDCLK) | J1-P3 (Test) |
|  | J1-P4 (DBG-RX)\* |
|  | J1-P5 (DBG-TX)\* |
|  | J1-P6 (MSP-TX) |
| J3-P11 (UART\_TXD) | J1-P7 (MSP-RX) |
| J3-P20 (GND) | J1-P8 (GND) |

#### “write\_it.bat”

The “AfridevV2ImageBuilder” folder has the batch file “write\_it.bat” to initiate the Flash programming of the Factory Manufacturing Test Image. This has the side effect of clearing the previous Factory GPS/Water/Modem test results.

When running the Factory code, the Debug Trace can be monitored using a tool like TeraTerm. The terminal must be configured with the following parameters:

• Baud rate: 9600

• Data Bits: 8

• Parity: No

• Stopbits: 1

• Protocol: none

The debug port is connected using the harness cable.

|  |  |
| --- | --- |
| RS232 to TTL  Adapter | AfriDev Connector |
|  | J1-P1 (3.3V) |
|  | J1-P2 (nRST) |
|  | J1-P3 (Test) |
|  | J1-P4 (DBG-RX)\* |
|  | J1-P5 (DBG-TX)\* |
| Debug TXD | J1-P6 (MSP-TX) |
|  | J1-P7 (MSP-RX) |
| Debug GND | J1-P8 (GND) |

#### “laststep.bat”

The “AfridevV2ImageBuilder” folder has the batch file “laststep.bat” to capture the Factory Test results to a text file and to program the Afridev2 unit’s flash with the production code. This process will not change or clear the Factory Test Results.

Should the terminal remain connected during programming the factory image, the Debug trace will display unintelligible characters. They will indicate that the firmware is communicating with the Modem, but the data will not be decipherable.

### OTA Update

A board that is already programmed with Afridev2 code can be update over-the-air with an AfriDev2 OTA command. The data that is within the AfridevV2\_MSP430\_msg.txt file can be sent to the device over the air, and the unit will store the new image in Flash, reboot and install this software.

The CRC of the file must be correct for the OTA Update to be committed. The update is processed in the usual Daily Update Time that occurs every 24 hours early in the morning GMT time.

# New and modified Functions

This is the first release of the Afridev2 Firmware. Comparing this firmware to the Original Afridev product, the following features have been added:

### New Water Sensing Algorithm

A new algorithm was developed that independently tracks the presence of Air or Water before 6 sensor pads. The new algorithm monitors “jumps” in capacitance values to detect water. When the values sharply jump lower, then there is water before the pad. Conversely, when the capacitance value jumps to a higher value, then there is air before the pad.  
  
Continuous Error Checking

The new water sensing algorithm has continuous error detection enabled by default. If water is detected to continuously pour for more than 5 minutes without stopping, the code will send a sensor data report and recalibrate the settings for air and water. If one or more pads are in an unknown state (when a pad detects water above a pad that detects air), the new algorithm can be set to automatically send a sensor data report and recalibrate the settings. In both cases, the sensor data report will give an idea of what is happening to aid in debugging efforts.

### Pad Temperature Tracking

### A part of the new algorithm requires the tracking of the air temperature just above the sensing pads. When the air temperature increases even a tenth of a degree, it causes the capacitance level to decrease towards a water detection. Likewise, a decrease in air temperature will cause the capacitance level to increase.

### To help avoid the false detection of water, the target capacitance levels for air and water are adjusted to track with the current temperature.

### The airflow within the housing is restricted, so the pad temperature will change based on the temperature of the board itself. As such, fast changes in ambient temperature may cause “unknown” reports (reports of higher pads seeing water when lower pads see air). Unknowns are not reported as water flow.

### Remote Water Sensing Control

### The code can process OTA requests to control the operation of the Water Sensing Feature.

### These operations assist in the investigation of Water Sensing Algorithm issues:

### Read Sensor Data: This tells the unit to broadcast the current sensing data as well as the “air” baseline data. The baseline data is only significant after a restart of the firmware (in case there is water on the pads when the unit restarted). The operation of the detection occurs without calibration needed.

### Overwrite Factory Data: This tells the unit to read the current pad values and record them as new “air” baseline data.

### Reset Water Detection: This tells the unit to re-acquire “water” and “air” target data. This could be a workaround if the unit reports frequent unknowns.

### Set Unknown Limit: This tells the unit to Reset Water Detection if “N” Unknown detections happen consecutively. This is disabled by default.

### Report Water Flow Data Now: This causes the unit to report every session when water is pumped. After a water session ends, water detection is disabled, the Modem is turned on and SENSOR DATA is reported, including the Total Liters for the pumping session.

### Set Downspout Rate: The water detection is tuned to a specific pour spout design. If the pump design is different in some way, this could cause different measurements. This value can be used in some cases to adjust the flow calculations.

### GPS Locating

### When the unit first powers up, the unit is “activated” after 50 liters or more of water is seen. Upon activation the Modem is turned off and Water Detection is stopped to measure the unit’s location. Upon finding a fix (between 1 to 5 minutes), the data is reported over the Modem.

### Watchdog Monitor

### The new Firmware has a Watchdog Monitor feature that will reset the unit’s Firmware if the hardware’s watchdog is not “hit” on a regular basis. This way, if a major failure of the Firmware occurs, the unit will restart itself. Bootloader Status Indicator The new firmware uses the LEDs on the board to show the status of the Bootloader. If all is well the Green LED will flash 3 times just before starting the application code. This would be followed with the Modem diagnostics flashing pattern (green/red alternately while the Modem is connecting) and a solid Red LED for 5 minutes if a connection was not sucessful, or a solid Green LED for 5 minutes if a connection was successful. If the Bootloader detects that the unit's flash does not have a CRC verified application loaded, then Both LEDs are lit and the unit waits in SOS mode. The unit will send an SOS message every 12 hours until a firmware update is received. If the Bootloader detects other errors before jumping to the Application code, then it will flash the Red LED a number of times to indicate what happened: List of Boot Loader Status Indications (each flash is 200 msec long with a 200 msec gap)

|  |  |  |
| --- | --- | --- |
| LED Sequence | Error Name | Description |
| 3 Green Flashes | Success | The Bootloader completed successfully |
| 2 Red Flashes | NMIIFG | The processor reports in the IFG1 register that it reset with a Non-Maskable Interrupt |
| 3 Red Flashes | OFIFG | The processor reports in the IFG1 register that it reset with a an Oscillator Failure. Sometimes this also causes the 3 reed beeps to flash quickly. |
| 4 Red Flashes | WDTIFG | The processor reports in the IFG1 register that it reset with a Watchdog Timeout |
| 5 Red Flashes | CRC Fail | The firmware image that was broadcasted to the unit does not have a correct CRC. It is not used |
| 6 Red Flashes | App Record Fail | The application code records an App Record in flah when the unit successfully operates successfully after a boot. The boot loader counts how many times the unit boots without application success, if this exceeds 64 times, then this signal is given and the unit enters SOS mode.. |
| 7 Red Flashes | Bad IVT | Corrupted Interrupt Vector Table, very bad. Enter SOS mode |

# Fixed Issues

From Version 2.2:

The water Flow Calculation code had a filter added to adjust for water that was pooling at the base of the sensor that was causing false detection of water volume. When only 1 pad is covered with water, the water volume is reduced to a tenth of what the Flow code was reporting. This adjustment and adjusting the Downspout Rate to 580 ml/sec yields an error rate of -6% error.

From Version 2.3

The water flow calculation code had a bug when the mean capacitance calculation equaled the water target number. In this case, the code was assigning a percentage of 0, instead of the 100 percent figure that was required. A term in the equation was using the remainder of dividing a term by 100, this limited the result to be up to 99. The new code looks for this border condition and assigns the value 100.

The downspout rate was adjusted for the final foam (6mm) and the detection code was measuring with a 2% error.

Code was added to watch the length of time that the unit reports the flowing of water. If water is seen for 5 minutes without stopping, it will be treated as a “Water Stuck” error and the water detect software is reinitialized. New values for air and water will be measured stopping the “stuck” condition.

From Version 2.4

A problem was observed in manufacturing tests that some of the units were freezing in the reset state. Investigating the firmware, there was a change made to the boot code’s reset vectors that was not applied to the application code.

BodyTrace and Twisthink requested that the opcode of the SOS message be changed because older products use the same opcode with a different message format. The opcode was changed from 0x06 to 0x23.  
  
From Version 2.5  
When a battery is inserted into the board, it is common that the connection does not go in right away and the unit re-starts up to 4 times until the connector is latched. A new board does not have an App Record in flash yet (see Bootloader Error Indicator section above) so the number of resets is counted in this state. The 2.5 Boot Loader entered SOS mode after 4 resets, which is too strict a test given the battery insertion resets, so the limit was changed to a much bigger number 64.  
  
From Version 2.6

The Version 2.5 code provided a workaround for a failure seen during Manufacturing Testing. After studying the TI specifications and articles on "Oscillator Failures" on the Internet, we found a requirement that was missed in the boot code.  
  
TI's MSP430G2955 MIXED SIGNAL MICROCONTROLLER specification, SLAS800, March 2013 p. 25 states in Note (2) that:

|  |
| --- |
| *(2) During power up, the CPU begins code execution following a period of* *td(BOR) after VCC = V(B\_IT-) + Vhys(B\_IT-) . The default DCO settings*  *must not be changed until VCC ≥ VCC(min), where VCC(min) is the minimum supply voltage for the desired operating frequency."* |

The specification states that td(BOR) is 2 mSec. The code, up to now, was not adhering to this requirement. In this version, a delay of 25 mSec was added before the processor's DCO settings to ensure compliance to this requirement.

From Version 2.7

The initial version of the code was setup with sensitive tolerances to report “water stuck” conditions if water were detected for 5 minutes without a break. After looking at live field data, this tolerance needs to be increased to avoid sending diagnostic information on higher volumes of pump activity. For this next version, the tolerance was increased to 30 minutes.

Additionally, there is another diagnostic that was disabled initially regarding “unknown states”. This version will cause a diagnostic report if the sensor reports an unknown state on 90 or more consecutive 2 second periods (3 minutes).

These changes will continue the effort of validating the accuracy of the sensors in the field.

# Known Issues

All known issues have been addressed.