

**Firmware Release 3.32**

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](#_Toc12626788)

[Firmware Build Directions 2](#_Toc12626789)

[Programming Methods 3](#_Toc12626790)

[New and modified Functions 4](#_Toc12626791)

[Fixed Issues 6](#_Toc12626792)

[Known Issues 7](#_Toc12626793)

.

# 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**. Separately install the CCS and Compiler images and then select the compiler version in the build file’s properties.

The release flash images are combined using a python script with the components “pip” and ”crcmod“ added. The version of Python used is **3.7.3 [MSC v.1916 32 bit (Intel)] on win32].** The latest version of pip is installed by entering the command-line command “python -m pip install -U pip”. Once installed, the crcmod component can be installed by entering the “pip install crcmod” command.

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 the ability to perform continuous error detection, but these features are disabled by default.

If enabled, and water is detected to continuously pour for more than 30 minutes without stopping, the code will send a sensor data report and recalibrate the settings for air and water.

If enabled, and 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.

### Set Wake Time: By default, the unit will enter low-power mode after 30 minutes of “dry operation” (no water seen). This setting has a half-second resolution (3600 = 30 minutes)

### 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 |

### Low Power Operation

### The Afridev2 firmware has been updated to go into low power mode when the unit has not seen water for 30 minutes. The unit will sleep for 20 seconds and then do a quick poll of the sensors for water, if water is seen then the unit goes back to full power mode.

Standing Water Detection

The Afridev2 firmware has been updated to recognize when there is water is standing in the base of the well head. It calculates the amount of the pouring volume that is attributed to standing water. Any 2 second period with a pouring volume less than this level will not be counted as pumped water. Should the amount of standing water change, a new higher level is established. The standing water detection is reset every time the highest pad is covered with water.

# Fixed Issues

Release (rev 2.7, May 30 2019)

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.

Release (rev 2.8, June 4 2019)

Added power saving mode that will put the Afridev2 into ultra-low power mode when it has not seen water for 30 minutes. The unit will sleep for 20 seconds and then poll the sensors for the presence of water. If the lowest pad detects water, then the unit will wake up and resume operations. While the unit is sleeping, no water detection is performed. When the unit is active communicating with the Modem or the GPS module, the low power mode is disabled.

Water detection was tuned to be more accurate. A new Down Spout Factor of 275 was tested with multiple 20 liter pours and the total pour values were verified to be accurate.

A filter to eliminate “trickle” data was added. When the bottom pad is covered with less than 50% water, then this data is not included in the estimate

Release (rev 3.1, July 3 2019)

Found root cause of 200 uA sleep current. The ADC that measures the internal pad temperature continued to run after the first reading of temperature. The new code stops the ADC module and allows the unit to reach the lowest sleep current level.

Release (rev 3.2, July 15, 2019)

In testing we saw that the unit stopped capsense measurements after 5 minutes. It was determined that in place of the sleep mode, David put in a “High Frequency”/”Low Frequency” sampling mode. This code needs to be removed as it is no longer needed.

Release (rev 3.3, July 24, 2019)

Updated a diagnostic feature called “water stuck” detection. The code originally reported a SensorData message if the unit continually reported water present for more than 30 minutes. For Rev 3.3, this interval was increased to 12 hours.

Release (rev 3.5, Aug 20, 2019)

Updated water sense code to detect standing water in the well head and to not count this water in the measured data. Added code to report trickle volume (the amount standing water that is not counted) in the SensorData and debug trace data.

Added code to measure 1 sample of water data after sleeping 20 seconds in low power mode.

Release (rev 3.7, Aug 30, 2019)

Updated code to detect standing water in the well and not report it as flow data. It was verified that standing water will not keep the unit from sleeping with inactivity.

Release (rev 3.8, Sep 13, 2019)

Increased time after pumping session to adjust air target from 4 seconds to 8.

Added 1 second RTC time every five 20 second sleeps to adjust for 1% error.

Release (rev 3.9, Sep 22, 2019)

Added debug information to observe the accuracy of the Software RTC at wake time and on Minute by Minute updates. Added NOP operation (0x08) to the Sensor Request (0x0f) OTA command

Release (rev 3.10, Sep 28, 2019)

Added a fine adjustment to the Software RTC. For every 1020 seconds sleeping, adding 3 seconds to clock. This yields an RTC with 0.02% error due to the sleep operation which amounts to 17 seconds a day.

Release (rev 3.11, Oct 3, 2019)

Added current RTC time to the response message for the Sensor Request (0x0f) OTA command with the NOP operation (0x08) request. The time will be in a binary format: hour\*256+minutes

Release (rev 3.12, Oct 19 2019)

Added code to synchronize all system timers with regards to sleep time corrections and one iteration of the sysexec loop that was missing from the time calculation on the software RTC.

Release (rev 3.13, Oct 31 2019)

Adjusted correction values for RTC accuracy when the unit is sleeping. The units were consistently sending hourly messages 180 seconds early, so the ratios were changed to yield 180 less seconds to add each hour.

Release (rev 3.14, Nov 1 2019)

To prevent Water Stuck conditions at startup, I removed the initialization code that sets starting air and water targets based on Manufacturing data. The temperature difference between the factory and the installed units may be great, and the thermistor measuring temperatures at the pads may have a lag due to the insulation properties of the plastic housing. This step was originally “insurance” in case the unit was reset or started when water was being pumped. At the moment there is a theory that this check may cause more problems then what it is worth. It is better to remove complexity in this case.

Release (rev 3.15, Nov 12 2019)

Adjusted RTC coming out of sleep mode according to live data from a few units

Added code to detect the growth of the margin between the air and water targets that will affect the accuracy of water measurements and cause “Water Stuck” sensor pads.

Release (rev 3.16, Nov 27 2019)

Disabled all testing features. Production release candidate.

Release (rev 3.17, Dec 14 2019)

Set the water stuck detection to 5 minutes and changed the code so that it handles the water stuck condition silently. This helps the unit adapt to sediment measurement volatility

Release (rev 3.18, Dec 20 2019)

Fixed a bug where the storage clock was advanced an hour before the previous hour’s water data was recorded.

Release (rev 3.19, Jan 31, 2020)

Fixed clock drift and alignment issues.

Release (rev 3.20, Feb 5, 2020)

Put back sleep drift adjustment.

Release (rev 3.21, Feb 19, 2020)

Removed version 3.15 margin calculation code, as it did not contribute to sediment data improvement.

Release (rev 3.22, Feb 25, 2020)

Adjusted gmtClockSetting to instantly change the storage clock to the same time. This way the activated and gps messages are produced at the expected time of 1:05am gmt, and not 24 hours after the gmtClockUpdate message.

Release (rev 3.23, Mar 13, 2020)

Trigger a red flag only when the current water measurement is 0, excluding when the current day’s 4-week average equals 0

Release (rev 3.24, Apr 25, 2020)

Modified sleep code to trigger within the timer ISR on a one second boundary. This is to reduce variability of the RTC.

Release (rev 3.25, Apr 25, 2020)

Test Release only. Enabled Hourly Timestamps

Release (rev 3.26, May 1, 2020)

Test Release only. Enabled Hourly Timestamps. Added code to clear hourly water reports in the first Water Data message that were not measured

Release (rev 3.27, May 3, 2020)

Test Release only. Removed storage clock alignment code from legacy software, Setting Storage Clock to RTC. Timestamp tests show that the storage clock is now synchronized with the RTC.

Release (rev 3.28, May 3, 2020)

Added code to re-sync the storage clock to the RTC when the modem is turned off.

Release (rev 3.29, June 26, 2020)

Added code to re-sync the storage clock to the RTC when the day’s water data is committed to flash. This eliminates the time drift. This is a test release that reports water data in 3 days instead of 7 to accelerate testing.

Release (rev 3.30, July 1, 2020)

The previous test release was set to 3 days reporting, but the product code was not able to process this interval. A bug caused thousands of daily water reports to be built unnecessarily. The test is back to a reporting interval of 7 days, which we know works.

Release (rev 3.31, July 1, 2020)

Before rolling out the v3.30 test, I reviewed the code some more and found that a recent change in v3.26 may have been the cause of the repeated water reports. There was a check for a backwards change of time that added 60 minutes to the storage clock, this would cause the next hour’s data to be immediately sent. I am scrapping the v3.30 test and I made this build to test this change instead.

Release (rev 3.32, July 5, 2020)

The last two releases exhibited a problem with the building of a flood of water reports when the storage clock was synchronized. After running locally, I found that the original design of the storage clock had to be replaced. It was designed to be its own clock, and when the value was changed it broke code lower down in the stack. I re-wrote the storage clock code to use only the RTC value and to process hourly and daily events when the RTC reached a certain level, no longer when the storage clock counted the seconds, minutes and hours itself. This takes out the complexity when the software sleeps 20 seconds at a time.

# Known Issues

Through data collection it has been determined that changing levels of sediment in the water is outside the capabilities of the Afridev2 sensor which uses Capacitive Sensing. There is no known way of detecting this condition looking at the capacitive measurements alone.