

October 2008 ModHopper Firmware changes.

Firmware updates for the ModHopper have a number of substantial changes. Users should be aware of these changes and how these may impact any custom applications that may have been developed. Although great care has been taken to maintain the previous feature set, users should test the firmware update prior to deploying this firmware on production equipment.

The firmware update will be provided at no cost.

Firmware release for version v2.04b
For R9120 and R24120, rev A and C boards.

Overview of changes:

This firmware update includes all changes from the v1.xx to v2.xx series to support the Rev-C hardware platform. A number of minor bug fixes are also included in this update for both Rev-A and Rev-C hardware.

Firmware Changes in v2.04b, Released 2008-10-23

- Changed dip switch read procedure to use a longer settle time and slower read cycle..
- Changed RSSI barchart function to handle all radio module types.
- Updated radio diags to report AU and other modules in the initial device setup program.
- Modified flag for SN radio to a bitmap field to allow multiple options.
- Fixed bug in pulse write-to-EEPROM procedure. Clear dirty flag first, then re-set it if the write fails. This prevents fast pulses from setting the dirty flag too late and skipping the next write if no more pulses come in.
- Fixed problem with pulse scanning to scan only once per ms, not 4x per ms.
- Added tool for strlcat to the project. (only the header is really needed, the .c file is for completeness)
- Added support for -5 radio firmware/sn/hw version numbers to device id table.
- Added support for new version numbers to identify some of the radio modules.

Firmware Changes in v2.03b, Released 2008-08-28

- Added point six dual analog transmitter type to the sensor data handler. This should also handle BAPI branded transmitters for temp, temp/rh and outdoor-temp.
- Updated -5 radio device init procedure to set up all the commands for operation..
- Added -5 radio to the project. Radio boot procedure now locates -5 module and has separate init procedure to set up features specific to that module.

Firmware Changes in v2.02b, Released 2008-06-27

- Changed 32 to 64 bit storage for serial number in sensor data packets.
- Added faster guardtime timeout to command to enter/exit command mode on -5 radio.
- Fixed point six counter/temp-xl to handle generic analog as temp-f as per instructions from Tim/Pointsix.
- Modified RSSI procedure to use xstream radio pwm output for RSSI calculation.
- Added conditional defines for larger block sizes of queue, sensor record store for rev C hardware
- Moved several init procedures out of main, and into library files.
- Fixed some bugs in the new packet decoder for the Counter/Temp-XL device.
- Updated system storage structure to include flag for linx radio installed. Updated id string to include SN in device p/n for linx module.



- Changed eeprom/fram nvmemory map. Old system (v1.xx) stored pulse count records in smaller blocks (16 byte offsets) New map is now using 20 byte records to hold count and runtime. When upgrading from older 1.xx firmware on rev-A systems, the user configured options for nv parameters and pulse count totals may be cleared.
- Created separate NV memory tables for Rev A vs. Rev C
- Fixed a bug in the runtime accumulator that counted runtime of open rather than closed contacts.
- Fixed a bug that allowed local 485 devices to be learned when the device is not in master mode, Fixed a bug that prevented 485 device route quality from being decremented when in slave mode and probing is suppressed.
- Fixed bug in signal strength handler to clear bargraph when no new data has arrived within 3 seconds.
- Added feature to configure Modbus port protocol mode using dip switch options on startup.
- Updated sample times for faster pulse rate options, added minimum dutycycle notes to the comments for each option.
- Added interrupt routine to check 485 TX status and turn off 485 TX direction pin (P1.25) for Rev C PCB.
- Added feature to detect RS485 direction control in hardware for rev-A and software in rev-C and set up the appropriate handler for the type of port found.
- Added packet counters for pointsix receiver on 232 port.
- Added a number of new Modbus registers to the point list for runtime/status points, linx radio stats, and config options. See documentation / Modbus register list for complete details.
- Added support to count good/bad packets from linx/pointsix transmitters.
- Fixed shutdown irq change. Do not ack irq so we don't get called over again.
- Updated eeprom/fram backup handler to deal with turning off i2c on power fail, also updated timing on fram and eeprom for specific hardware rev A and C.
- Added procedure to detect pulse count closures using a/d converter to measure input resistance.
- Added support for contact closure runtime counting on pulse inputs terminals.
- Updated pulse count procedure to implement pulse rate option.
- Updated Modbus registers to allow write options for pulse speed and threshold.
- Modified default for contact closure to 500ohms.
- Added option for pulse count speed setting.
- Added procedure to measure RSSI using the radio module PWM feature.
- Added feature to store best RSSI value while receiving a packet on the linx radio. Updated min/max off-time filter to weed out bogus packets.
- Added features to save rssi for onboard linx receiver in data packets. Fixed byte order problem with receiver processor for fast-counter module.
- Internal ADC on RevC hardware measures system voltage and temperature.
- Updated Modbus registers to include voltage and pcb temp.
- Fixed a bug in bootcount register reporting.
- Added register for contact closure threshold.
- Cleaned up buffer processing procedure for linx packets. Added support for 400mhz led upon receiving packets on the linx radio. (rev-c only)
- Added set/clear/disable/blink function for the 400Mhz led on the rev-c hardware.
- Updated point six data packet processing. Now able to store and report data packets from point6 modules correctly.
- Changed command for point6 data point processing to use binary data function.
- Updated Modbus ping procedure to avoid pings when in point six mode.
- Added test to 485 port byte processing to terminate on CR mark in point six mode.
- Added a hex digit to integer conversion function.
- Updated prefs management to handle protocol mode of 4 for point-six receivers.
- Added feature to require at least 2 packets from a single sensor serial number be received before data is cached. Should prevent spurious received data from showing up.
- More work to simplify the point-six receiver packet processing.
- Added shell of function to receive data from the point-six wireless receiver module (pointview LR)
- Updated Badger/Orion system to terminate packet receiving and process it when a LF mark shows up in the data stream. Updated interpacket timeout to 10ms.
- Added IRQ compiler tag to the shutdown irq handler.



- Added assembly segment to timer interrupt handler to allow other IRQs to interrupt this handler. This allows for irq nesting as per AN 10381, and is required to let the linx receiver irq to work in a timely manner.
- Updated irq definition to include "IRQ" type for gcc. moved timer re-set register to start of function.
- Changed record storage to purge old historical records, not new records that have not been downloaded.
- Enable the A/D converter in the power control register for rev-c pcb.
- More work on packet storage procedures. Added preference feature to configure the minimum time between 400mhz packets to store, Fixed packet storage cleaning procedure to clean old records properly.
- Fixed led_on(), led_off(), and led_stat() to correctly handle LED_ALIVE. Alive LED pin (P1.18) must be driven low to turn LED off, but allowed to float high to enable LED.
- Set up orion software to transmit chars when in orion 232port mode to keep the 232 converter powered.
- Updated i2c addresses to use conditional defines for rev A vs C. Rev A is on eeprom starting at address 0 thru 128 Rev C is on fram starting at address 512 thru 1023
- Changed compiled versions for different processor models on the rev A and C hardware.
- Added error response for Modbus packet query registers out of range
- More work making the firmware report the sensor records correctly
- Updated dipswitch handler procedures for rev A and C differences.
- Working data storage, with support for 400Mhz and Orion data packets.
- Added Orion source file to receive and parse the packets from the Orion receiver.
- Updated preferences to store Orion serial port mode. Updated prefs defaults for Orion serial port mode and high speed pulse options.
- More work on making the ModHopper report packet data from 400mhz/badger devices.
- Added Modbus register for pulse counting speed option.
- Started adding memory handling procedure to store wireless sensor records from devices like Point Six and Badger Meter systems.
- Removed polling of pulse inputs from radio interface library, the interrupt handler version should deal with this now.
- Implemented new pulse counter and backup scheme. Pulses are counted in interrupt. Data is written to eeprom from mainline code. Shutdown irq now only sets a flag. pulse backup storage now has room for runtime values.
- converted procedures to use virtualized i2c data addresses rather than using chip/bank/offset style notation.
- Moved interrupt enable/disable into lpctools.c/h for other code files to use. Added i2c bus clock defines and sanity check.
- Copied over i2ceeprom source files from A8812. Started updating lib to use virtual addresses.
- Updated boot sequence to save boot status code for use in a Modbus register. The older v1.xx series firmware reported only 0x01=watchdog-reset. The new firmware v2.xx and later reports all LPC boot codes in the same format as the A8812 onboard io.
- Added a Radio_Shutdown function and corresponding hardware specific handlers. This function is intended to be called from a power fail interrupt, so it must be interrupt safe.
- Updated code to use timer.c/h in separate file copied from A8812 codebase.
- Updated startup code to use undef handlers, run in supervisor mode. matches a8812 start sequence.
- Added watchdog.c/h separate source file from a8812 project.
- Added new boot codes for undefined handlers, including SpuriousHandler.
- Added MAM setup codes to boot sequence
- Changed stats counter to only count 485 timeout when the address is a known valid 485 address.
- Added Modbus register feature for ForceMaster mode. Increased allowed 485 timeout to 5 seconds.
- changed Modbus address procedure to send a route delete packet with our old address prior to changing to the new Modbus address.
- Updated Modbus id handler to allow for different radio types.
- Added feature to get timeout as a function to allow ease of use with different radio types.

Note: Revised Sep 16, 2010 with new company street address.

