Ultra-low-power, long-range Arduino communications

This remote sensor module can operate for around one month on a single lithium button cell and transmits data to a remote station at least 500m away. Standard stub antennas are good for at least 500m line-of-sight while the home-made BiQuad antenna shown in the photo extends the range to at least 1500m, with a gain of around 12dB. Depending on how well the transmitter and receiver antennas are aligned, even greater ranges are possible.

The transmitter uses about 6-7mA while active, however it only needs to be active for a second or two periodically to transmit data. The average current is much lower, at around 0.3mA if transmitting data every 30-40 seconds. This equates to 6mAh/day, so a 150mAh cell will last around 25 days, while a Panasonic 220mAh CR2032 cell will last 30 days.

Both the transmitter and receiver units are based on an Atmel ATmega328PU microcontroller, as used in the Arduino Uno. They run at 4MHz and are connected to NRF24L01+-based 2.4GHz 2Mb/s low-power radio transceiver modules with a built-in power amplifier (for transmit) and a low-noise amplifier (for receive). The sensor unit also has a DHT22 sensor to monitor relative humidity and ambient temperature.

Because Arduino is relatively easy to program, constructors could quite easily alter the software to support more and different sensors but the DHT22 serves as a useful example of the capability of this design.

The radio modules can operate in one of 126 different channels, numbered 0-125 and spanning 2400MHz to 2525MHz, ie, each channel is separated by 1MHz. In Australia, channels 0-83 are able to be used however they are likely to include a lot of WiFi and Bluetooth traffic. Unfortunately, channels 100-125 overlap the 2.5GHz band which is licensed for Electronic News Gathering and use of the remaining channels is not permitted.

Perhaps the best strategy is to choose channel 0 or 83 as these are right at the edges of the WiFi spectrum and likely to have lower interference. The radio modules can also be configured for different data rates, CRC lengths and power levels; in this application, we're using a 256Kbps data rate, 16-bit CRC and maximum power (PA_MAX, 20dBm or around 100mW).

Between data bursts, the radio module is put into a low-power standby mode and the microcontroller is in "sleep" mode. Only its watchdog timer is left running and this wakes it up periodically, to gather data from the sensor and then transmit this via radio to the receiver station.

Software

The software to control the micro's sleep state is in the Arduino library file "lowpower.h". The NRF24L01 library includes a powerDown() function to put the radio module into its sleep mode. This library communicates with the radio module using an SPI interface. The micro also switches off power to the DHT22 sensor when it's not being used, as it's powered from one of the general purpose outputs.

Note that an Arduino Uno normally uses a 16MHz crystal so you need to flash the blank chip with the 8MHz Arduino bootloader (which uses the internal oscillator instead) before uploading the sketch. Note also that the circuit cannot be powered from 5V as the radio module's maximum supply voltage is 3.3V nominal.

There isn't much to the transmitter circuit. Besides the cell, power

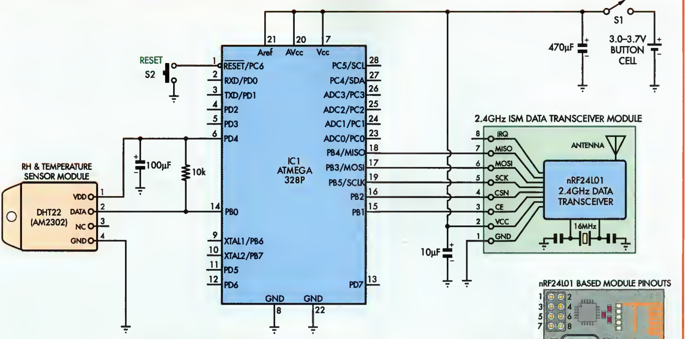
switch, micro, sensor and radio module, the only other components are three supply bypass capacitors, a pull-up resistor for the DHT22 open-collector output, plus a reset switch for the micro. The large supply bypass for the DHT22 is necessary because the I/O pin driving it can't supply a lot of current, so the capacitor is charged before the sensor is queried to avoid its supply voltage dropping too much during operation.

The receiver circuit is similarly spartan, with the DHT22 sensor removed and a 128x64 graphic LCD fitted instead, which is driven from the micro over an I^2C serial bus. Received data is also sent to the Arduino's serial console and may be logged on a PC for later analysis.

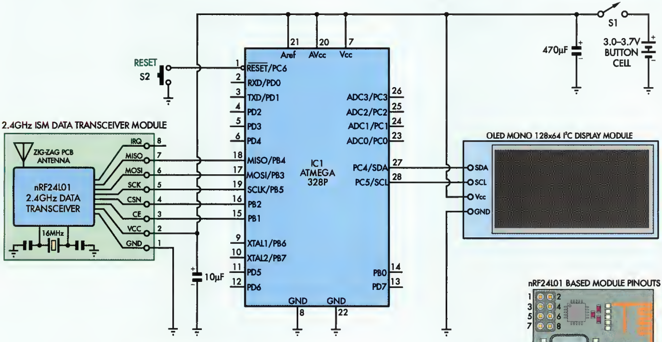
While the receiver is shown powered from a button cell, it draws a lot more current than the transmitter unit (partly due to the LCD), so a mains-derived 3.3V regulated DC supply may be more practical. Take care that the supply voltage doesn't exceed the specified 3.7V maximum.

Antenna

For shorter ranges, you can use standard 2.4GHz stub antennas which connect directly to the SMA socket on the radio modules and are available at low cost. However, much better range can be achieved using a home-made bi-quad antenna. The ideal dimensions for your chosen channel can be determined from



SENSOR/TRANSMITTER CIRCUIT



RECEIVER/DISPLAY CIRCUIT

this website: http://buildyourownantenna.blogspot.in/2014/07/double-biquad-antenna-calculator.html

Simply plug in the frequency you're using (eg, 2400MHz for channel 0) and press the "Calculate" button. Then form the antenna from stiff wire of the specified diameter, which will be around 1.38mm. This equates to a cross-sectional area of 6mm. It's soldered to the back of a panel-mount RF connector which is then attached to a sheet of aluminium, copper or copper laminate of the specified dimensions, around 125x125mm.

The photo shows one of the prototype antennas which achieved excellent performance at a distance of 1.5km. It wasn't possible to test at further ranges as no convenient location with line-of-site was available; it's possible that these antennas will provide much longer range than that.

This project is quite cheap to build, with the radio modules, sensors and LCD costing just a few dollars each on eBay or Ali Express. The software for both units (*LowPowerArduinoLongRangeComms.zip*) can be downloaded from the SILICON CHIP website.

Somnath Bera,
Vindhyanagar, India. ($70)