This draft manuscripts provides an overview of the SDI12-wsn project hosted at <https://github.com/jkennedy-usgs/SDI12-wsn>. THE MANUSCRIPT HAS NOT BEEN REVIEWED BY USGS AND IS NOT AN OFFICIAL PUBLICATION. It is provided for informational purpose only.

**Adaptation of the SDI-12 protocol for easy deployment earth science wireless sensor networks**

**Jeffrey Kennedy,** U.S. Geological Survey

**James Wagner,** Oregon Research Electronics

**Keywords:** wireless sensor network, soil moisture, data collection,

**Abstract**

The SDI-12 protocol, used for data loggers to communicate with hard-wired digital sensors, has been adapted to facilitate the deployment of wireless sensor networks by means of a hardware/firmware bridge unit that relays communication between an SDI-12 enabled data logger and a wireless sensor network. The fundamental benefit is that existing data-collection infrastructure can be readily expanded to take advantage of wireless sensor networking. Unlike other available SDI-12 products, which enable point-to-point wireless communications with SDI-12 enabled sensors, the current effort provides for deployment of low-cost mesh topology, sleeping wireless networks that accommodate any analog-output sensor. The concept is extendable to any particular wireless sensor networking protocol, and is demonstrated using the DigiMesh protocol, 50 mW 900 MHz radios, and dielectric permittivity soil moisture sensors. Although the sleeping nature of wireless sensor networks prevents true implementation of the SDI-12 protocol, whereby sampling is initiated by the host data logger, the system offers considerable potential for the expansion of existing data collection infrastructure.

**1. Introduction**

Wireless sensor networks—specifically, networks implementing “mesh” topology with low-power radios and “sleeping” to conserve power while not communicating—have long offered the potential for significant advances in the field of environmental data collection (for example, Rundel et al., 2009; Porter, 2005). Although notable protocols (ZigBee, those by Campbell Scientific, Inc., or Memsic, Inc., for example) and deployments (for example, Delin et al., 2005; Barrenetxea, 2008; Ritsema, 2009) have been developed, wide-spread adoption is still hindered by significant challenges in setting up the networks. For example, the ZigBee protocol requires establishing each radio in advance as an end node or router, requiring advance knowledge of where each one will be placed in the field. Real-time operating systems with wireless networking capability, such as TinyOS, implemented in commercially-available hardware such as the MEMSIC Mica2 motes, offer great flexibility at the expense of complexity, and require fairly advanced programming to deploy. Other wireless sensor network (WSN) products, such as the MEMSIC eKo motes, are designed for easy deployment, but must be deployed as a integrated system and are not compatible with existing data collection infrastructure.

To overcome difficulties in deploying WSNs, and to take advantage of existing data collection infrastructure, we have developed a hardware/firmware bridge unit that bridges SDI-12 and WSN communciation. SDI-12 is a ubiquitous protocol found on thousands of data loggers worldwide, designed to allow multiple (wired) sensors on a single bus. The system described can be deployed with no greater difficulty than any other SDI-12 sensors. Both hardware and firmware are provided in the public domain, and are designed in such a way as to be easy to build and deploy. Furthermore, the cost of the hardware is low – about $80 USD per remote radio, and cheaper if produced in quantity. The online supplement to this article contains source code, printed circuit board (PCB) layouts, and a bill of materials.

WSNs are ideally suited to soil moisture measurement, and many successful implementations have been demonstrated (for example, Bogena et al., 2010; Ritsema et al., 2009; Cano et al., 2007). A primary reason is the low power requirements of typical soil moisture probes – typically 10’s of mA only during excitation and measurement. Furthermore, soil moisture at depths deeper than within a few cm of the surface varies slowly, meaning nodes can be sleeping much of the time, saving power. Finally, soil moisture demonstrates significant spatial variation, particularly during dry periods (Western et al., 1999) and many distributed measurements are helpful to adequately characterize an area. In the United States, soil moisture is a spatially under-sampled component of the hydrologic cycle (a map showing current soil moisture measurement sites is at *http://az.water.usgs.gov/projects/soilmoisture*). Rainfall is well-characterized by the National Weather Service’s NEXRAD Doppler radar sites, and streamflow is well-characterized by over 7,000 gaging stations operated by the U.S. Geological Survey. Several satellite missions have made great progress towards remote soil moisture sensing, but still require adequate ground truthing to ensure accuracy (such as the recent CanEx-SM10 experiment). By making it easy to add distributed soil moisture measurements to existing data collection sites, the SDI-12 to WSN bridge presented here can help increase the spatial distribution of soil moisture data, with benefits to agriculture, flood forecasting, drought monitoring, and other fields.

**2. The SDI-12 protocol**

SDI-12 (Serial Data Interface – 1200 baud, *www.sdi-12.org*) is a protocol developed by the U.S. Geological Survey and private industry to allow digital communication between a host data logger and sensors. Compared to analog sensors, this allows longer cable distances between data logger and sensor. Other advantages are that multiple sensors, up to 50 or more, can be deployed at a single data logger, and sensors can do calculations, such as converting voltage or current measurements to useable data such air pressure, within the sensor. SDI-12 ports are found on nearly every data logger sold for earth science data acquisition in the past 10 years; the USGS alone maintains over 7000 data loggers at gaging stations, each with an SDI-12 port. This ubiquity, together with the fact that nearly all currently deployed SDI-12 data loggers are part of existing data collection networks, where data is relayed to a central repository via radio, cell-phone, or satellite telemetry, or collected manually on a regular basis, makes the SDI-12 an ideal candidate for the incorporation of WSN.

The basic SDI-12 protocol involves a host data logger initiating measurement at a specific sensor. Each sensor has an address set using a hardware switch, or via SDI-12 commands. Upon receiving a command to start measurement, the sensor replies with the number of seconds until a measurement is complete, and the number of measurements that will be returned. Upon completion of this period, the data logger issues a request for data, to which the sensor replies with one or more measurements. Alternatively, the sensor may respond with a service request before the previously specified elapsed time, upon which the data logger issues a request for data. Other SDI-12 commands expand upon this basic protocol, such as allowing concurrent measurements to be made at multiple sensors, or requiring the sensor to transmit a checksum to verify data integrity. Non-measurement SDI-12 commands are used to set sensor addresses, retrieve information (such as manufacturer or sensor type) from a particular sensor, or query the SDI-12 bus to see which sensor addresses are active.

The SDI-12 protocol mandates inverted, 0-5V logic levels on a single bidirectional bus. In addition, SDI-12 data loggers provide +12V for sensor power (if needed) and ground.

A fundamental limitation of integrating SDI-12 with a sleeping WSN is that while nodes are asleep, they are unresponsive to incoming messages (this could be remedied by using a WSN protocol incorporating “wake on receive” functionality, with a concomitant increase in power consumption). In the current design, the network sleeps for fifteen minutes between soil moisture measurements, making it unrealistic to expect the host data logger to wait until the network is awake before a measurement is returned. Therefore, the WSN is sampled each time it wakes, and data are stored in the bridge unit memory. When an SDI-12 command for a measurement is received, the bridge unit returns the average of stored, previously sampled data. This is not present a serious limitation when sampling a slowly varying property such as soil moisture. For more rapidly varying phenomena, the sleep time can be shortened as appropriate, keeping in mind a more robust power supply may be needed. An additional consideration when deploying at data loggers with existing telemetry is the increased bandwidth needed for the additional WSN data. The present design stores 16 soil moisture samples at 15-minute intervals, to be retrieved by the SDI-12 data logger at 4 hour intervals. Sleep and wake times, and the number of samples to store in memory from which to compute the average, are readily set by modifying the source code but are not intended to be changed during deployment.

**3. Wireless Sensor Networks**

The basic concept of a WSN is a group of sensors, or nodes, deployed over a geographic area to collect data about a spatially distributed condition. Furthermore, it is typically understood to use some form of relaying of messages, or routing, rather than point-to-point communication between a base station and remote node. WSN hardware is usually designed to be inexpensive, with low-power radios relative to the size of the network, and low-power, implementing some type of “sleeping” algorithm whereby nodes enter a reduced functionality state to save power. More advanced WSNs, termed sensor webs (Delin, 2002), are differentiated by the ability to process data at each node and share information with other nodes, thereby increasing the ability to respond to transient environmental phenomena. Sensor webs have proven successful in hydrology-related deployments (Delin et al., 2005), but no commercial hardware is readily available.

WSN protocols are rapidly evolving. To date, the only protocol with any significant adoption among vendors is the ZigBee protocol (*www.zigbee.org*), an extension of the IEEE 802.15.4 standard. Although ZigBee is available free of charge to non-commercial users, it is not open source and commercial users are required to join the ZigBee Alliance. ZigBee-based projects have been successfully deployed in the earth sciences (for example, Zhu et al., 2005; Bogena et al., 2007).

One alternative to the ZigBee protocol is Digi, Incorporated’s proprietary DigiMesh protocol, implemented in the work presented here. DigiMesh offers several advantages not available with ZigBee. First, every node unit can serve as either an end node or router, and does not need to be configured as such before the network is deployed. Routers can collect data from attached sensors and sleep in coordination with the end nodes. Second, using the ad-hoc on-demand distance routing method, DigiMesh offers self-configuration and self-healing, in which node units automatically determine the most efficient radio links between them, and, if a node unit becomes unresponsive, the network can determine an alternative route. This offers obvious advantages for maintaining continuity of data collection in a remote environment. Finally, the sleeping protocol in DigiMesh is self-synchronizing and is easy to implement. In time, an open source alternative to DigiMesh will likely become available. At present, however, we feel that the advantages outweigh the disadvantages of using this proprietary protocol. Because we implement DigiMesh without modification or access to the internal communications, this project remains in the public domain.

The DigiMesh protocol is implemented using an API framework. Rather than sending data to the node unit directly, API mode encapsulates messages and commands in a packet with ancillary information such as packet number, destination address, event notifications, etc. This adds functionality—in particular, the ability to retrieve digital and analog I/O data from remote nodes directly without needing a separate microcontroller at each node.

**4. Design Implementation**

The principal consideration in designing the system is ease of deployment. To achieve this, the system is intentionally of limited flexibility—it samples at a predefined interval and accommodates only a single variation of sensor (single-ended analog output). The result is that no programming or other special skills are required.

**4.1 Hardware**

The hardware design comprises two parts: the bridge unit that physically connects to the SDI-12 host data logger and the remote node units (Figure XX). Each unit requires one radio. The test design utilizes XBee Pro 900 MHz radios, but the design is readily adaptable to other XBee radios and frequencies. PCB layouts for the bridge and node units—included in the online supplement—are readily manufactured at low cost, particularly in volume, by any circuit board manufacturer. The PCB design has also been uploaded to the BatchPCB service and can be ordered directly (http://batchpcb.com/index.php/Products/XXXXX). Where possible, all components are through-hole for ease of assembly. The size of the bridge unit could be reduced if needed by incorporating more surface-mount components.

4.1.1 Bridge unit

The bridge unit comprises 4 main systems: voltage regulation, ESD protection, and level-shifting/buffering/inverting circuits; an Atmel AVR microcontroller; an XBee radio; and an LCD for communicating status messages to the user. No external inputs—buttons or switches—are used or needed.

Both a 5V and 3.3V voltage regulators are needed: the former provides the proper voltage for SDI-12 communication and the latter powers the AVR, XBee, and LCD. Both regulators use 12V input from the SDI-12 bus. ESD protection is as per the SDI-12 protocol. Level shifting, inverting, and tri-stating on the SDI-12 bus are controlled by two ICs. The CMOS SN74AHCT1G125 serves as a tri-state buffer, controlled by an output from the microcontroller. TTL-compatible, it also shifts the 3.3V Tx line from the AVR to 5V. The SN74LVC2GU04 dual inverter gate inverts the AVR UART to match the SDI-12 inverted logic on both the Rx and Tx lines, and shifts the 5V SDI-12 bus to 3.3V.

The Atmel AVR microcontroller used is the ATMEGA644PA. The primary considerations for selecting this IC were dual UARTS, for communication with both the SDI-12 bus and the XBee radio; sufficient I/O to support the LCD and other necessary ports; availability of DIP packaging for ease of assembly; and a JTAG (in-circuit debugging) interface. Other microcontrollers, from AVR or other manufacturers, would be equally suitable. The hardware design implements both SPI and JTAG programming interfaces. A 16 MHz crystal is used; slower frequencies would likely work as well but have not been tested. A reset button provides hard-reset capability. Because power is provided by the host data logger and is presumably robust, power-saving features of the microcontroller have not been implemented; power consumption is relatively low in comparison to the 80 mA consumed by the bridge unit XBee.

The XBee Pro 900 communicates with the AVR via one of the UARTS. Three LED’s connected to the XBee indicate signal strength (RSSI), sleeping status of the network, and network association (lit to indicate the XBee is communicating with others in the network). The XBee mounts in a 2mm pitch socket. A reset button provides hard-reset capability.

An LCD is connected via 4-pin parallel interface to port A of the AVR. The unit is designed to use a 3.3V LCD with a ST 7036 controller, although other units could be easily implemented. No backlight has been implemented but one could easily be added.

4.1.2 Node unit

The node unit is essentially a carrier board for the XBee radio. ADC and digital I/O is handled by the XBee’s onboard AVR microcontroller, as implemented in the Digimesh firmware. ADC is 10-bit, providing accuracy to about 0.0012 V using a 1.25 V voltage reference. The SDI-12 address for each node is set in binary through a DIP switch connected to four of the XBee’s digital I/O, an acceptable solution for addresses 0 through 9 and A through F; higher addresses would require an alternative. A hardware reset switch and “commissioning” button, for facilitating joining the XBee to an existing network, are also included. Probes are connected to the XBee using two, three-position screw terminals.

Power is provided by 2 AA batteries and a 6-cm square solar panel. In initial testing this has been sufficient to provide power through a three day cloudy period; the node unit will run for approximately 5 days on AA batteries alone at the 15 minute sample interval used. XBee Rx (awake, not transmitting) and Tx power requirements are 80 mA and 210 mA, respectively, at 3.3V; standby power is 60 µA. Power regulation is provided by a Maxim MAX856 IC, which is 80% efficient between about 1 mA and 100 mA at a supply voltage between 1.5V and 3.3 V. Efficiency drops to 60% during the short periods when the XBee is transmitting. The MAX856 is capable of sourcing 3.3V down to an input voltage of 0.8 V. It also supplies a 1.25 V voltage reference to the XBee ADC, accurate within 0.8%.

Utilizing a 2 dBi 6” duck antenna, XBee Pro 900 radios have a line-of-sight communication distance of 2 km, and an indoor-urban distance of 140 m, sufficient for nearly all deployments where spatial variability of soil moisture is the variable of interest. Lower-power, cheaper XBee variants could likely be substituted (with possibly some firmware changes). Alternatively, higher gain antennae, including multi-element Yagi up to 15 dBi, are approved. The antenna used in the node unit attaches directly to the XBee. A short extension is used on the bridge unit so that the antenna may be mounted outside the host data logger enclosure.

The node unit enclosure is epoxied to a 45-degree PVC junction, which in turn mounts at the top of a 2 m, 2.54 cm diameter PVC post (fig. field\_deployment). Probe cables are routed through the post to minimize the potential for disturbance, and using PVC minimizes the potential for RF interference. Burial of the post to a depth of 0.5 m is sufficient for stability.

4.1.3 Soil moisture probes

The probes used are Decagon, Inc. 10-HS soil moisture sensors. Any other soil moisture probe, or other sensor, that outputs 0 to 1.25 V will be directly compatible with the node unit hardware, and other analog sensors could be used with minimal modification. In response to a 3 to 12 V excitation, the 10-HS outputs a 0.3 to 1.25 V signal, independent of excitation voltage, which is proportional to apparent dielectric permittivity by a fourth-order polynomial. Because of the large permittivity contrast between water and air, volumetric water content can be derived using the well-known Topp equation (Topp and others, 1980), a third-order polynomial. Because the polynomial for determining permittivity is probe-specific, it is not programmed directly into the bridge unit, and must be applied separately either by the host data logger or as a data-processing step.

**4.2 Firmware**

Firmware consists of a collection of c-language libraries compiled using avr-gcc, a port of the GNU c compiler for AVR microcontrollers. It should be readily adaptable to other platforms and compilers. Program architecture is based around two state machine frameworks, one each for the WSN and the SDI-12 interface. These have been kept independent, so that the same firmware may be adapted to WSN protocols other than DigiMesh, or the SDI-12 code adapted for non-WSN projects. Commented code is available in the online supplement to this article.

The WSN state machine is implemented in main.c. The initialize and interrupt service request (ISR) routines within main.c contain XBee specific code; otherwise, all XBee routines are in wireless\_xbee.c. Generic XBee routines, not specific to this implementation, are in xbee\_API.c. WSN initialization under DigiMesh involves broadcasting a “node discovery command” to all node units, upon which each one responds in random order with a unique identifier. Once these IDs have been compiled, the initialize routine steps through each node, reading the SDI-12 address assigned on the DIP switch. From this point on, each node unit is referred to internally by the SDI-12 address. These addresses need not be sequential. Memory is reserved for the maximum possible number of nodes, about 20 in the 64k flash memory of the ATMEGA644PA (after space is used for code). The initialize routine also sets the XBee I/O settings (inputs, outputs, and pullup resistors) and writes them to non-volatile memory.

Following initialization, the firmware enters the WSN state machine (fig. 3). Each time a valid XBee packet is received, the UART Rx ISR sets the state to WSN\_message\_waiting, from which wireless\_parse\_message is called and sets the appropriate state, depending on the content of the packet. Information about each node, including the XBee address, data from the attached sensors, and communication errors, is stored in the \_node struct, and all of these structs are stored in the nodes array. Array node\_ids is a lookup table for SDI-12 addresses, where value at the sequential array index (0 through the number of nodes) is the (non-sequential) SDI-12 address. That is, node\_ids[0] is the SDI-12 address of the first node found during the node discovery process (which could be 0-9, A-Z, etc.), node\_ids[1] is the SDI-12 address of the second node found, and so on.

Requests for data from the SDI-12 data logger are handled by an independent state machine (fig. 4) and the UART Rx ISR on the second AVR UART (which operates independently of the UART used for WSN communication). Because baud rates (1200 and 9600, for SDI-12 and XBee, respectively) are slow relative to the AVR clock (16 MHz), both UARTs can receive and process incoming data simultaneously. SDI-12 code executes in response to Rx or Tx interrupts; pin change interrupt on the SDI-12 bus, timer interrupt, or when the WSN state machine calls sdi12\_dotask. The SDI-12 state machine indicates that a request for data has been received by setting a flag (sdi12\_msg\_signal) equal to the SDI-12 address requested. When this flag is checked by the WSN state machine, the SDI-12 response is prepared, and a pointer set to the memory location of the response. When sdi12\_dotask is called, the SDI-12 state machine replies to the host data logger using a Tx interrupt.

**References**

Barrenetxea, G., Ingelrest, F., Schaefer, G., Vetterli, M., Couach, O., Parlange, M., 2008. SensorScope: out-of-the-box environmental monitoring, Proceedings 2008 International Conference on Information Processing in Sensor Networks, pp. 332-343.

Bogena, H.R., Huisman, J.A., Oberdörster, C., Vereecken, H., 2007. Evaluation of a low-cost soil water content sensor for wireless network applicatioons. Journal of Hydrology 344, 32–42.

Bogena, H. R., Herbst, M., Huisman, J. A., Rosenbaum, U., Weuthen, A., Vereecken, H., 2010. **Potential of Wireless Sensor Networks for Measuring Soil Water Content Variability.** Vadose Zone Journal 9(4), 1002-1013.

Cano, A.,  Lopez-Baeza, E.,  Anon, J.L., Reig, C., Millan-Scheiding, C., 2007. Wireless Sensor Network for Soil Moisture Applications. In: Proceedings International Conference on Sensor Technologies and Applications, Valencia, Spain, pp. 508 – 513. doi 10.1109/SENSORCOMM.2007.4394971.

Delin, K.A., 200. The Sensor Web: A Macro-Instrument for Coordinated Sensing. Sensors2, 270-285.

Delin, K.A., Jackson, S.P., Johnson, D.W., Burleigh, S.C., Woodrow, R.W., McAuley, J.M., Dohm, J.M., Ip, F., Ferré, P.A., Rucker, D.F., Baker, V.R., 2005. Environmental Studies with the Sensor Web: Principles and Practice. Sensors 5, 103-117.

Porter, J., Arzberger, P., Braun, H.-W., Bryant, P., Gage, S., Hansen, T., Hanson, P., Lin, C.-C., Lin, F.-P., Kratz. T., Michener, W., Shapiro, S., Williams, T., 2005, Wireless sensor networks for ecology, BioScience 55(7), 561-572.

Ritsema, C. J., Kuipers, H., Kleiboer, L., van den Elsen, E., Oostindie, K., Wesseling, J. G., Wolthuis, J.‐W., Havinga, P., 2009. A new wireless underground network system for continuous monitoring of soil water contents. Water Resources Research 45, W00D36. doi:10.1029/2008WR007071.

Rundel, P.W., Graham, E.A., Allen, M.F., Fisher, J.C., Harmon, T.C., 2009. Environmental sensor networks in ecological research, New Phytologist 182, 589–607. doi: 10.1111/j.1469-8137.2009.02811.x.

Topp G.C., Davis, J.L., Annan, A.P., 1980. Electromagnetic determination of soil water content: measurements in coaxial transmission lines. Water Resources Research 16, 574–582.

Western, A.W., Grayson, R.B., Blöschl, G., Willgoose, G.R., McMahon, T.A., 1999. Observed spatial organization of soil moisture and its relation to terrain indices. Water Resources Research 35(3), 797-810. doi:10.1029/1998WR900065

Zhu, Y.W., Zhong, X.X., Shi, J.F., 2006. The design of wireless sensor network system based on ZigBee technology for greenhouse. Journal of Physics: Conference Series 48, 1195–1199. doi: 10.1088/1742-6596/48/1/223.