

Wenn diese einzelnen Clients oder Knoten ein Maschennetz bilden, bei dem jeder Knoten mindestens einen Nachbarn hat, der für ihn das Weiterreichen von Daten übernehmen kann, wird dies als »vermaschtes Netz«, in Englisch als MESH, bezeichnet. Hierfür ist eine Software (Meshrouting) notwendig, die den besten Weg für das Weiterleiten der Daten zum jeweiligen Ziel entsprechend anpasst, sobald Netzwerkknoten ausfallen, neue hinzukommen oder Knoten bewegt werden.

Bei WLANs, die mit einem Access Point (AP) ausgestattet sind, findet die Kommunikation nicht direkt zwischen den Knoten, sondern stets über den AP statt, was als Infrastrukturmodus bezeichnet wird. Durch die Verwendung eines oder auch mehrerer Access Points lässt sich eine Verbindung von WLAN-Clients (Mobile Units) zu einem üblichen LAN schaffen, und auch die Kopplung von separaten LANs ist über Access Points möglich.

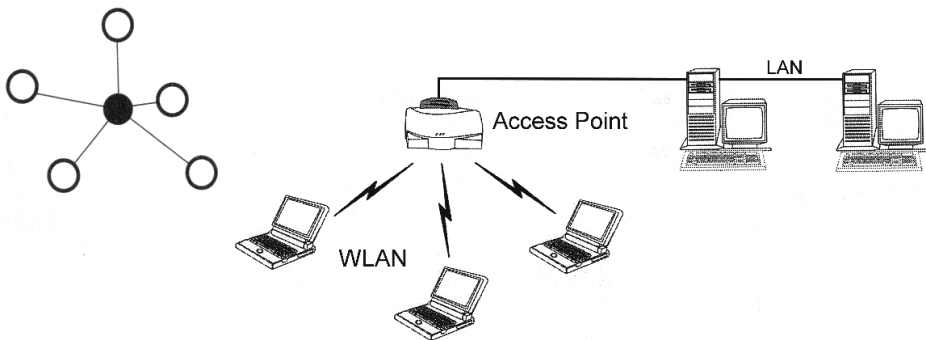


Abb. 6-5 WLAN-Realisierung mit Access Point (Infrastrukturmodus), der an ein LAN angeschlossen ist.

6.5 Sensorknoten im WLAN

Weil die WLAN-Adapter in erster Linie für hohe Datenübertragungsraten konzipiert sind, ist ihr Energieverbrauch relativ hoch, sodass sie sich – zumindest auf den ersten Blick – nicht für den Aufbau von stromsparenden Sensornetzen eignen, bei denen die Sensorknoten möglicherweise mit Energy-Harvesting-Methoden (siehe Abschnitt 5.4) versorgt werden sollen.

Wie bei allen Wireless-Knoten hängt es ganz wesentlich von der Firmware und dem möglicherweise selbst geschriebenen Programmcode sowie letztendlich von der Anwendung ab, wie hoch der Leistungsbedarf jeweils ausfällt. Für den Aufbau eigener Schaltungen werden entsprechende Wi-Fi-Transceiver als einzelne Chips angeboten, etwa von den Firmen Maxim Integrated

(MAX2828/2829) oder Broadcom (BCM4360), was demnach noch einen recht hohen Entwicklungsaufwand nach sich zieht.

Einfacher geht es mit fertig aufgebauten Wi-Fi-Modulen für Embedded Systems, wie sie von Microchip (MRF24WB0), H&D Wireless (SPB105) oder Sagrad (SG901) angeboten werden. Diese Module werden meist per SPI oder auch über eine andere Schnittstelle wie I²C oder RS232 mit einem Mikrocontroller verbunden.

Für die Ausführung des TCP/IP-Stacks, wovon es auch Versionen als Open Source gibt, ist ein entsprechend leistungsfähiger Mikrocontroller notwendig. Microchip empfiehlt hierfür die eigenen PIC-Mikrocontroller, H&D Wireless einen AVR32-Mikrocontroller (EVK1105-Kit) und Sagrad einen ARM-Mikrocontroller.

Der Stromverbrauch hängt nur bedingt vom jeweiligen IEEE 802.11-Mode ab, weil die maximale Sendeleistung bei Standards mit höheren möglichen Datenraten gegenüber denen mit geringeren Datenraten typischerweise reduziert ist, was letztendlich in der Hardware des Chips begründet liegt. Prinzipiell ist dank einer höheren Datenrate weniger Energie notwendig, um eine bestimmte Menge an Informationen zu übertragen.

Das Modul WSN802 der Firma RF Monolithics (RFM) wird explizit als *Wireless Sensor Network Module* bezeichnet. Die Modulplatine verfügt insgesamt über 30 Anschlüsse, die über zwei Steckerleisten im 1,27-mm-Raster herausgeführt sind. Hier sind die Verbindungen für eine Antenne, ein *Serial Peripheral Interface* (SPI), zwei 10-Bit-A/D-Wandlereingänge, einen 16-Bit-PWM-Ausgang und vier digitale I/O-Ports zu finden.

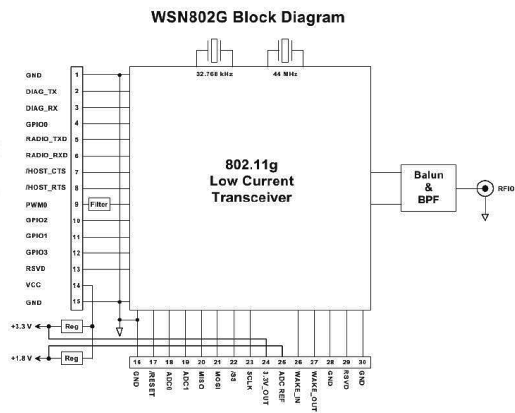
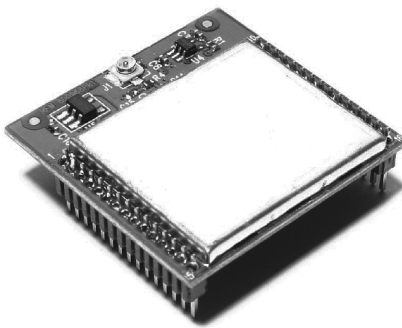


Abb. 6-6 Das WSN802-Modul verfügt über 30 herausgeführte Anschlüsse für die Verbindung mit Peripherieschaltungen.

Der Transceiver unterstützt maximal den IEEE 802.11b-Modus (11 MBit/s) und ist aufgrund der IEEE-Standardisierung gleichermaßen in neueren IEEE 802.11-Netzen (g, n) einsetzbar. Bei einer Sendeleistung von 10 dBm wird ein Strom von ca. 200 mA bei einer Versorgungsspannung von 3,6 V verbraucht, was mit den oben erwähnten IEEE 802.11-Modulen vergleichbar ist und ungefähr um den Faktor 10 höher liegt als bei ZigBee- oder anderen Low-Power-Modulen. Im Sleep Mode werden lediglich 7,5 μ A verbraucht, und auch im Schlafmodus bleibt das Modul weiterhin Mitglied im Netzwerk.

6.5.1 Inbetriebnahme

Zum WSN802-Modul gibt es ein Entwicklungskit mit allen benötigten Komponenten und einer Software für die Konfiguration des Moduls sowie zur Durchführung von Testmessungen, sodass die erste Inbetriebnahme meist nur wenige Minuten in Anspruch nimmt. Auf dem Board sind zwei I/O-Ports (GPIO) mit Tastern verdrahtet, und an den beiden A/D-Wandlereingängen sind ein Potenziometer und ein Temperatursensor angeschlossen.

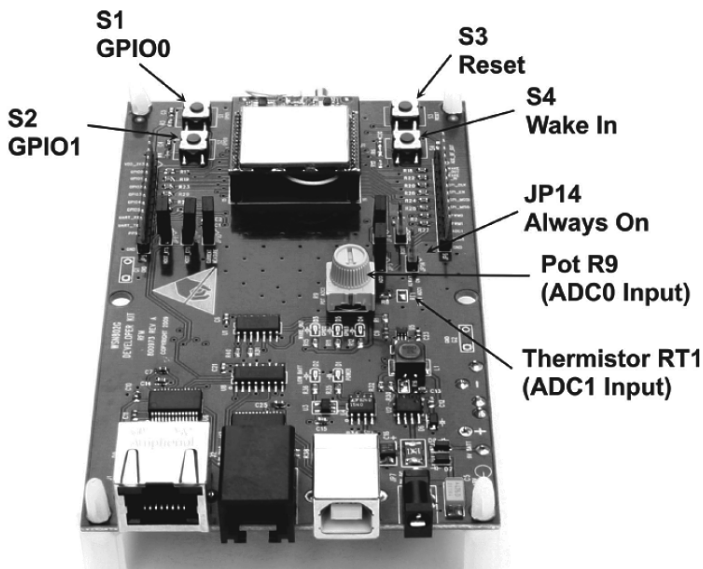


Abb. 6-7 Das WSN 802-Modul ist auf ein Entwicklungsboard gesteckt und kann hiermit leicht konfiguriert und erprobt werden.

Das Modul ist von Haus aus so konfiguriert, dass es in einem einstellbaren Intervall aus dem Sleep-Modus aufwacht, die konfigurierbaren Eingänge abfragt,

diese Daten per WLAN verschickt und dann wieder selbsttätig in den Sleep-Modus schaltet.

Für eine WLAN-Testumgebung ist neben dem Evaluation Board ein WLAN-Access-Point nötig, dem eine WSN-SSID, ein WSN-Passwort sowie eine Verschlüsselungsmethode (WPA/WPA2) zuzuteilen sind. Die Modulkonfiguration kann mit dem Evaluation Board über einen LAN-(RJ45-) oder über einen USB-Anschluss oder auch per WLAN erfolgen, was mit dem Programm WSNConfig.exe auf einem PC ausgeführt wird.

Sind die Netzwerkdaten für alle drei WLAN-Einheiten (Modul, Access Point, Client) entsprechend konfiguriert, verbindet sich das Modul automatisch mit dem WLAN. Es wird im Programm WSNConfig.exe angezeigt (siehe Abbildung 6–9) und ist bereit für die weitere Konfigurierung (Messzyklus, Sleep-Modus etc.). Standardmäßig werden alle 20 Sekunden die gemessenen Werte des Potenziometers, der Temperatur und sowie die Pegel der Taster per WLAN als UDP-Pakete auf dem Port 8255 übertragen und können auf dem Client-PC per WSNConfig.exe oder mit dem reinen Visualisierungstool WSNApp.exe angezeigt werden.

Das Modul kann grundsätzlich in zwei verschiedenen Modi betrieben werden: zum einen als Client in einem Netzwerk, der die Daten per WLAN an einen Server (PC) schickt, und zum anderen in einem Ad-hoc-Netzwerk, in dem das Modul selbst den Host darstellt. Damit ist es möglich, den Knoten auch in einer Umgebung zu betreiben, in der kein WLAN vorhanden ist, etwa als Host für ein Smartphone. Zu beachten ist allerdings, dass der Sensorknoten im Ad-hoc-Modus nicht energieeffizient betrieben werden kann. Das Modul versucht vorrangig, sich mit einem vorhandenen Netzwerk zu verbinden. Funktioniert dies nicht, zum Beispiel weil gerade kein entsprechendes WLAN vorhanden ist, schaltet das Modul selbsttätig in den Ad-hoc-Modus und verbleibt in diesem, bis es ausgeschaltet oder neu gestartet wird. Außerdem ist es nicht möglich, das Modul im Ad-hoc-Modus in den Sleep-Modus zu versetzen, weil es die WLAN-Verbindung sofort trennt, sobald es nicht mehr aktiv ist.

6.5.2 Systemaufbau

Vom Hersteller ist es nicht vorgesehen, dass der Controller vom Anwender (um-) programmiert werden kann, sodass diese Software quasi als unveränderliche Modul-Firmware anzusehen ist. Die Konfigurierung des Moduls ist standardmäßig ausschließlich mit dem PC-Programm WSNConfig.exe möglich, und die damit getätigten Einstellungen werden im Flash-Speicher des Moduls abgelegt.

Bei genauerer Betrachtung fällt auf, dass das WSN 802-Modul auf einem SoC (System on Chip) GS1011 (102 Pin, QFN-Gehäuse) der Firma GainSpan basiert, das in Deutschland bei der Firma Scantec erhältlich ist. GainSpan bietet hierfür ebenfalls ein passendes Evaluation Board (PB013) mit der dazugehörigen Firmware und Application-Software an. Die Abbildung 6–8 zeigt das Innenleben des SoC GS1011 und damit des WSN 802-Kerns, der demnach zwei ARM7-CPU's enthält. Eine CPU ist zuständig für das WLAN-Subsystem, die andere dient als Network-Prozessor und zur Anbindung der Peripherieschnittstellen (SPI, I²C, UART, GPIO usw.).

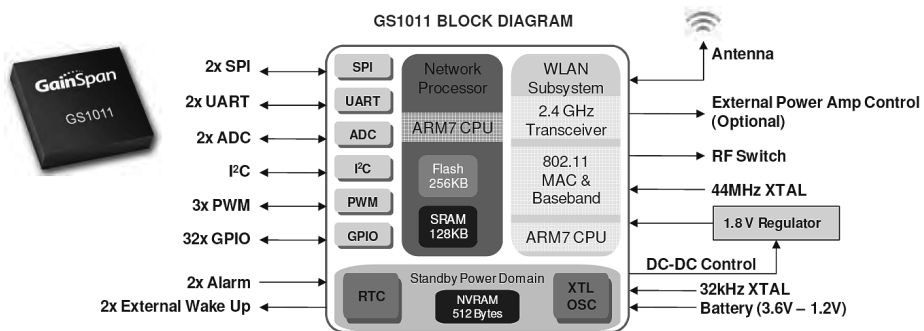


Abb. 6–8 Das SoC GS 1101 der Firma Gainspan beinhaltet zwei ARM7-CPU's.

Wie RFM geht auch Gainspan davon aus, dass der Anwender das SoC nicht nach seinen Wünschen umprogrammiert, sondern er ist lediglich zur Durchführung eines Firmware-Updates berechtigt, das von Gainspan bzw. RFM zur Verfügung gestellt wird. Zur Steuerung des SoCs bzw. das WSN 802-Moduls in einem Embedded System wird stattdessen das Hinzufügen eines separaten Mikrocontrollers vorgeschlagen, der über eine der Schnittstellen mit dem SoC kommuniziert und die Alarm- sowie Wake-Up-Leitungen steuert.

6.5.3 Konfigurierung

Das Modul benötigt als Versorgungsspannung 3 V bis maximal 3,6 V. Das Entwicklungsboard (siehe Abbildung 6–7) wird mit einer Spannung von 9 V versorgt, und ein Schaltregler (LM22671) setzt diese Spannung auf 3,3 V um. Naturgemäß verbraucht das Modul im Sleep-Modus am wenigsten Strom (7,5 μ A), sodass genau zu überlegen ist, wann und für welche Zeitspanne es aktiv zu schalten ist, um Eingänge oder Sensoren abzufragen und die Messwerte per WLAN zu übertragen.

Solange am WAKE_In-Pin des Moduls ein High-Pegel anliegt, bleibt es aktiv. Das heißt, um in den Sleep-Modus zu wechseln, ist dieser Pin lediglich auf Low-Pegel zu setzen. Zu beachten ist, dass es – je nach RFM-Firmware-Version – auch genau anders herum sein kann.

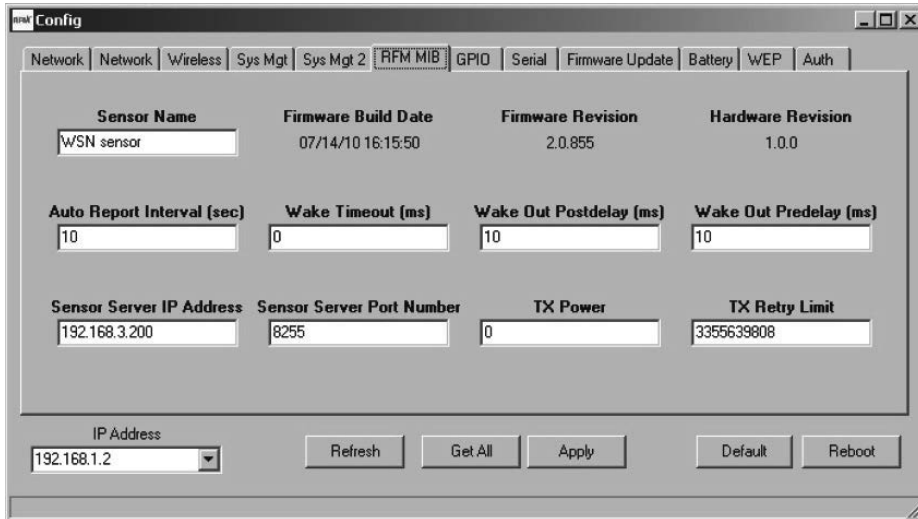


Abb. 6–9 Das WSN 802-Modul erlaubt eine Vielzahl von Einstellungsmöglichkeiten, wobei die RFM Management Information Blocks (MIB) von besonderer Bedeutung für die Applikation sind. Es ist hier die höchste Ausgangsleistung (0 = 10 mW) unter TX Power festgelegt.

Daneben gibt es weitere Methoden, um das WSN 802-Modul aufzuwecken bzw. aktiv zu halten, was über die Software (Abbildung 6–9) in den Konfigurationsabschnitten *Wireless* und *RFM MIB* einstellbar ist.

- **Auto Report Interval:** Nach einer in Sekunden festzulegenden Zeitspanne wird das Modul aus dem Sleep-Modus aktiv geschaltet.
- **LinkUp Trap Interval:** Diese Timer-Einstellung sorgt dafür, dass standardmäßig alle 60 Sekunden ein Signal ausgesendet wird, um die WLAN-Verbindung aufrechtzuerhalten. Dieser Wert soll laut RFM nicht über zwei Minuten hinausgehen. Der Timer kann im Gegensatz zu den anderen hier erwähnten Timern und der Funktion über den WAKE_In-Pin nicht abgeschaltet werden.
- **Config Trap Interval:** Wenn das hier anzugebende Intervall abgelaufen ist, das standardmäßig auf 10 Sekunden gesetzt ist, werden die Konfigurationsaufgaben übernommen, die mittels der Konfigurationssoftware bestimmt wurden. Nach der grundlegenden Konfigurierung des Moduls – wenn dann keine oder

nur noch selten Konfigurationsänderungen notwendig sein sollten – wird dieser Wert hochgesetzt, etwa auf einige Male pro Tag.

- **Wake Timeout:** Dieser Wert bestimmt, wie lange das Modul nach dem Aufwecken in Millisekunden aktiv bleibt, um auf eintreffende Befehle zu warten.
- **Wake Out Postdelay, Wake Out Predelay:** Diese beiden Delays bestimmen das Verhalten des Wake-Out-Anschlusses, nachdem das Modul aktiv wurde, indem jeweils ein Wert in Millisekunden für eine Zeitverzögerung *nach* oder *vor* der Aktivschaltung definiert wird.

Grundsätzlich verbleibt das Modul – quasi unabhängig von den Timer-Einstellungen – auch dann im aktiven Modus, wenn die IP-Adressen nicht korrekt gesetzt worden sind, momentan ein serielles Byte empfangen wird oder Daten per WLAN gesendet oder empfangen werden oder ein High am WAKE_IN-Pin anliegt. Am Anschluss WAKE_OUT wird stets dann ein High-Signal ausgegeben, wenn sich das Modul im aktiven Modus befindet, was zur externen Steuerung, typischerweise mit einem separatem Mikroprozessor, genutzt werden kann. Mithilfe der Optionen *Wake Out Predelay* und *Wake Out Postdelay* lassen sich hierfür Verzögerungen in Millisekunden festlegen, damit der Pegel an WAKE_OUT eine bestimmte Zeit vor (Predelay) oder nach (Postdelay) der Aktivzeit des Moduls geschaltet wird, was für Initialisierungs- und Abschaltphasen von Sensoren oder der Spannungsversorgung genutzt werden kann.

6.5.4 Erweiterungsschaltungen

Das Modul kann nicht komplett abgeschaltet, sondern lediglich in den stromsparenden Sleep-Modus geschickt werden. Für Low-Power- und typische Energy-Harvesting-Schaltungen ist das Trennen bestimmter Einheiten (Sensoren, Transceiver) von der Spannungsversorgung ein probates Mittel, den Stromverbrauch effektiv zu reduzieren. Hierfür ist als steuernde Instanz nicht immer ein Mikrocontroller notwendig. Der Einfachheit halber könnte die Spannungsversorgung des Moduls auch mithilfe eines Timers oder einer Echtzeituhr – über einen MOSFET – direkt abgeschaltet werden.

Der Stromverbrauch dieser zusätzlichen Elemente muss dabei (wesentlich) geringer sein als der Stromverbrauch des Moduls im Sleep-Modus (7,5 μA) selbst. Außerdem darf es nicht zu Problemen führen, wenn das Modul dann nicht mehr im WLAN als Client präsent ist. Ein gewöhnlicher Timer wie der NE555 benötigt selbst in der stromsparenderen CMOS-Version 80 μA im Standby, was demnach keine Lösung wäre.

Eine Echtzeituhr wie der Typ PCF2123 (NXP) ist da weitaus genügsamer und benötigt lediglich 100 nA für das »Weiterticken«. Mithilfe einer am Modul ange-

schlossenen *Real Time Clock* (RTC) kann das Modul zu den eingestellten Zeiten mit der Spannungsversorgung verbunden werden, sodass nur der minimale Stromverbrauch der RTC in den (Modul-)Passivzeiten ins Gewicht fällt. Anders als bei der Verwendung eines Mikrocontrollers muss eine RTC zwar nicht programmiert werden, gleichwohl muss sie gesetzt werden, was beim Typ PCF2123 per SPI erfolgen kann. Dafür wird Energie benötigt, die beispielsweise in einem Kondensator vorgehalten werden kann, der beim Verbinden mit SPI geladen wird, bevor der Harvester, etwa eine Solarzelle, in Aktion treten kann.

Die Firma Cymbet, die für ihre Lithium-Akkus (EnerChips) in Dünnschicht-technologie bekannt ist (siehe Abschnitt EnerChips auf Seite 197) hat eine Schaltung in Form eines USB-Sticks entwickelt, die einen EnerChip CBC3112 (3,3 V mit 12 μ Ah), eine RTC (RV-2123 von Micro Crystal) sowie ein USB-Interface (FT2232 von FTDI) enthält, was als Evaluation Kit *CBC-Eval-06* vertrieben wird. Dazu gehört eine Windows-Software, die das Stellen der RTC und das Setzen der Alarmfunktionen gestattet. Die RTC-Schaltung läuft nach dem Abziehen vom USB-Port weiter, indem sie vom Cymbet-Akku versorgt wird. Nach einer kleinen Modifizierung eignet sie sich als Echtzeituhr für die Steuerung des WSN802-Moduls.

Eine Solarzelle liefert, je nach Lichtquelle und -einfall, eine unterschiedliche Ausgangsspannung, sodass für den Betrieb einer Elektronik (Sensorknoten) eine Spannungsstabilisierung notwendig ist. Für das WSN-Modul ist eine Spannung von 3,3 V ideal, die von einem DC/DC-Konverter, wie etwa dem Typ MAX1705 (16 Pin QSOP) von der Firma Maxim Integrated, geliefert werden kann. Die genaue Schaltung hierfür ist in Abbildung 6–11 gezeigt, Abbildung 6–10 stellt das Blockschaltbild hierzu dar.

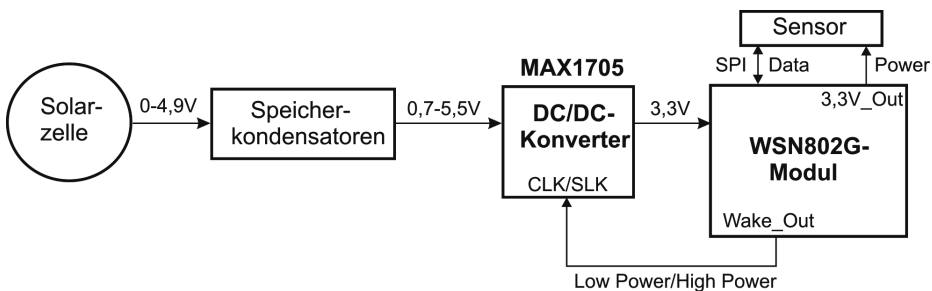


Abb. 6–10 Blockschaltbild für den Betrieb des WSN 802G-Moduls als Sensorknoten mit der Versorgung durch eine Solarzelle

Der verwendete Konverter entspricht einem Step-Up-Wandler mit Linearregler und liefert eine Ausgangsspannung im einstellbaren Bereich von 2,5 V bis 5,5 V,