

단일-홉 무선 센서 네트워크를 위한 고속 병렬 소프트웨어 업데이트 기법*

남영진*, 박영균*, 남인길*
*대구대학교 컴퓨터·IT공학부
e-mail: yjnam@daegu.ac.kr

A High-speed Concurrent Software Update Technique for Single-hop Wireless Sensor Networks

Young Jin Nam*, Young-Kyun Park*, In-Gil Nam*
*School of Computer and Information Technology, Daegu University

요 약

무선 센서 네트워크상에서 센서노드의 소프트웨어 변경, 패치 등의 작업을 위해서 센서노드별로 ISP와 같은 케이블을 이용하여 프로그램 해야 한다. 이는 네트워크 내에 존재하는 노드의 수가 소량인 경우 크게 문제되지 않지만 대량의 노드로 구성된 경우 프로그래밍을 위한 시간적인 측면과 투여되는 인력적인 측면에서 매우 높은 비용이 소요된다. 본 논문에서는 현재 많이 사용하고 있는 IEEE 802.15.4기반의 단일-홉 무선 센서 네트워크 환경에서 노드의 수와 무관하게 고속 병렬로 센서노드 내부의 프로그램을 무선으로 업데이트 할 수 있는 기법을 제시하고 실험을 통하여 그 성능을 검증한다.

1. 서론

근래의 무선 센서 네트워크는 생활의 편의성을 도모하기 위해 건물 내에 설치되는 경우도 있지만 목적에 따라 원격지나 사람이 접근하기 힘든 지역 혹은 위험한 지역의 정보를 수집하고 감시하기 위한 목적으로 이용되기도 한다. 경우에 따라서는 조밀한 지역에 대량의 노드가 설치될 수도 있고 비용을 고려해 최소 비용으로 최적의 구조로 설치되는 경우도 있다. 하지만 어느 경우이건 단순 배터리 교체에서부터 소프트웨어의 변경, 수정, 재설치 등 센서노드의 유지, 보수가 필요하다. 센서노드의 소프트웨어에 관련된 유지, 보수의 경우, 사람이 일일이 수거해서 다시 프로그램을 할 수 밖에 없는데 각각의 노드를 수거하는 과정에서부터 새로 프로그램 하는 과정까지 시간적, 인적 비용이 많이 들 수 밖에 없다. 특히, 관리해야 하는 센서노드의 수가 많아지면서 그 비용은 매우 심각할 정도로 문제가 된다.

이러한 문제점을 해결하기 위해 무선으로 센서노드 내에 포함된 소프트웨어를 효과적으로 업데이트하기 위한 다양한 연구[1-3]들이 진행되어 왔다. 각각의 기법마다 Atmega128L MCU를 포함한 다양한 종류의 센서노드를 사용하고 있으며, 본 논문에서는 특히 코드 뱅킹기능을 제공하는 8051 MCU를 이용하는 센서노드를 이용한다. 최근 대부분의 무선 센서 네트워크는 센서노드에 장착된 IEEE 802.15.4 표준을 지원하는 모듈을 이용하고 있으며, 본 표준의 제약성 때문에 단일-홉 무선 센서 네트워크를 일반적으로 구성하여 사용하고 있다. 표준에 따르면 단일-홉 내에는 이론적으로 최대 255개의 센서노드가 포함될 수 있으며, 무선을 이용하여 이들 센서노드 각각에 대한 소프트웨어 업데이트 작업이 필요로 한다. 단순히 순차적인 방

법으로 업데이트 작업을 진행할 경우, 모든 작업이 종료될 때 까지의 소요시간은 센서노드 수에 비례하게 된다. 본 논문에서는 소프트웨어를 무선으로 병렬로 업데이트 하는 기법을 제시함으로써, 무선 센서 네트워크 내에 존재하는 센서노드의 수와 무관하게 소프트웨어 업데이트를 위해 일정시간이 소요되도록 한다.

2. 관련연구 및 배경지식

2.1 기존 소프트웨어 업데이트 기법

소프트웨어 업데이트 기법은 전송하는 이미지의 변환 여부, 멀티 홉 지원 여부 등과 같은 특징에 따라 구분 지을 수 있다. 업데이트할 소프트웨어의 이미지 일부를 업데이트 하는 방법에는 표 1과 같이 Incremental[4,5], TinyCubus[6], Trickle[7]등이 있으며, 소프트웨어의 전체 이미지를 전송하는 기법에는 XNP[8], MNP[9], Deluge[10], MOAP[11], Infuse[12], Sprinkler[13], Aqueduct[14] 등이 있다. Deluge의 경우 TinyOS 2.x 로 코드가 공개되어 있지만 EEPROM이 SPI로 MCU와 연결되어 있고 MCU가 RFIC와 SPI버스를 공유하면 동작하지 않는 문제가 있다.

<표 1> 기존 무선 소프트웨어 업데이트 기법

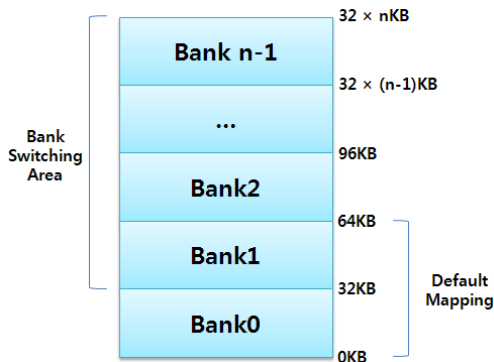
기법명	전송데이터	전송프로토콜	지원홉	업데이트범위
Incremental	Delta patch	--	Single hop	Whole Network
TinyCubus	Modular update	--	Multihop	Selected Nodes
Trickle	Mate script	adv-req-data	Multihop	Whole Network
XNP	Complete program	--	Single hop	Whole Network
MNP	Complete program	There phase (adv-req-data)	Multihop	Whole Network

* 본 연구는 교육과학기술부와 한국산업기술재단의 지역혁신인력양성사업의 연구결과로 수행되었음

Deluge	Complete program	There phase (adv-req-data)	Multihop	Whole Network
MOAP	Complete program	Ripple neighborhood-by-neighborhood publish	Multihop	Whole Network
Infuse	Complete program	parent/child (TDMA)	Multihop	Whole Network
Sprinkler	Complete program	parent/child (TDMA)	Multihop	Whole Network
Aqueduct	Complete program	--	Multihop	Selected Nodes

2.2 8051 MCU의 메모리 구조

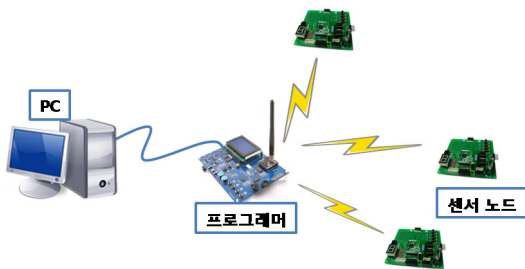
8051 호환 MCU는 메모리 어드레싱 영역이 64KB로 제한되어 있다. 하지만 프로그램 크기가 64KB를 초과할 경우 코드뱅크 이라는 기능을 사용할 수 있는데, 전체 내부 플래시 메모리를 어드레싱 가능한 크기인 32KB의 뱅크들로 나누고 프로그램 이미지를 각 뱅크에 분할하여 저장 [15]하고 필요에 따라 변경해가며 사용한다. 그림 1은 8051 MCU의 메모리 구조를 나타낸다. 코드뱅크 모드를 사용하게 되면 기본적으로 Bank0와 Bank1이 어드레싱 되도록 맵핑된다. Bank 0에는 Start Up코드, Bank Switching코드, main코드 및 ISR(Interrupt Service Routine)이 포함된 이미지가 저장되어야 한다. Bank 1에서 Bank n-1의 영역은 필요에 따라 수행할 코드가 저장된 Bank로 스위칭해서 사용하게 된다.



(그림 1) 8051 MCU의 메모리 구조

3. 제안기법

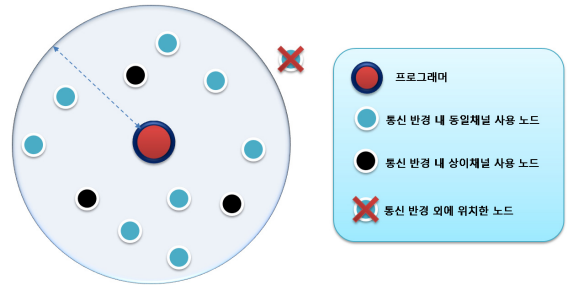
본 논문에서 제안하는 무선 센서 네트워크 상 소프트웨어 업데이트 기법은 크게 1) 각 노드의 EEPROM을 삭제하는 단계, 2) 업데이트 할 이미지를 전송하는 단계, 3) 데이터 오류 검사 및 센서노드 리셋 단계로 이루어진다.



(그림 2) 제안기법의 하드웨어 구성도

우선 업데이트 할 소프트웨어의 이미지파일을 센서노드로 전송해줄 노드가 필요한데, 이 노드를 '프로그램머'로

칭한다. 프로그래머는 PC와 UART로 인터페이스하며 센서노드의 외부 EEPROM의 삭제 명령, 체크섬(checksum)확인 및 리셋(reset)확인 명령, 프로그램 이미지파일의 패킷화 및 전송 기능을 담당한다.

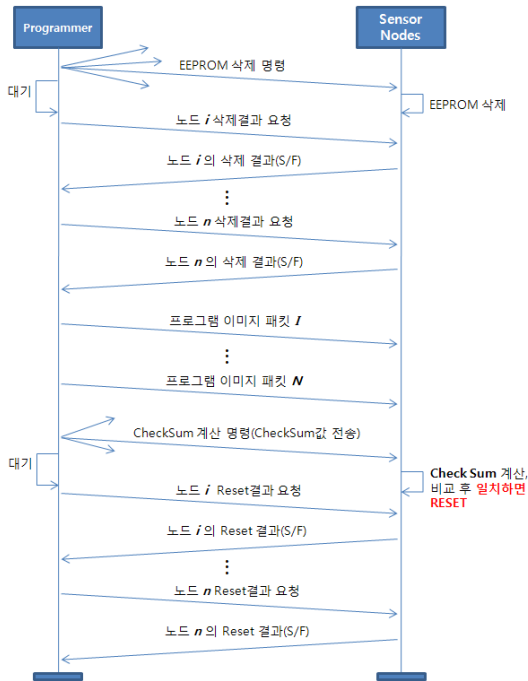


(그림 3) 제안기법 적용가능 노드의 범위

제안기법은 그림 3과 같이 프로그래머의 통신 반경 내의 동일 채널을 사용하는 모든 노드가 대상이 될 수 있으며, 각 센서노드는 사전에 소프트웨어 업데이트를 위한 프로토콜이 프로그램 돼있어야 한다. 그림 4는 제안하는 프로그래머와 센서노드간의 소프트웨어 업데이트 프로토콜 시퀀스를 나타낸다.

- 프로그래머는 EEPROM 삭제 명령을 브로드캐스트 한 후 센서노드가 EEPROM을 삭제하는데 걸리는 시간만큼 대기한다. 사용되는 8051 MCU와 I2C로 연결된 EEPROM의 크기는 64KB이며, 프로그래머는 삭제 명령 전송 후 실험적으로 구한 12초를 기다린다.
- 센서노드는 명령을 수신 후 EEPROM을 0xFF로 쓴다. 프로그래머는 EEPROM삭제 대기시간이 지난 후 유닉캐스트(즉, 점대점)로 각 노드와 삭제 완료/실패 여부를 확인한다. 이 작업은 상기 EEPROM 삭제시간에 비해서 매우 짧은 시간 내에 완료되며, 노드 수가 증가하더라도 그 증가량은 크지 않다.
- 이후 프로그래머는 업데이트 할 이미지를 IEEE 802.15.4 표준 패킷단위(순수 데이터 크기 64바이트)로 분할해서 모든 노드가 받도록 브로드캐스트 전송한다.
- 센서노드는 패킷 수신 후 프로그램 이미지 영역을 EEPROM에 차례대로 저장한다.
- 프로그래머가 전송한 이미지에 대한 체크섬을 담은 체크섬 계산 명령과 성공시 리셋 명령을 모든 노드에 브로드캐스트 전송하고 센서노드가 체크섬을 계산하는 시간과 리셋하는 시간 만큼 대기한다.
- 노드는 EEPROM에 저장된 이미지의 체크섬을 계산하고 수신한 값과 같으면 체크섬 계산 결과를 저장하는 전역변수를 초기화 한다. 이후 EEPROM의 이미지를 8051 MCU의 프로그램 메모리 영역으로 복사하고 watchdog 타이머를 이용해 스스로 리셋한다. 체크섬 값이 틀린 경우 체크섬 계산 결과를 저장하는 전역변수를 0xFF로 설정해 둔다.
- 프로그래머는 체크섬 계산 및 리셋 대기 시간이 지난 후 EEPROM 삭제 확인과 마찬가지로 각 센서노드와 1:1로 리셋 여부를 확인한다.
- 센서노드는 리셋 확인 명령을 받으면 체크섬 계산 결과를 저장해둔 전역변수를 반환함으로써 체크섬 계산 및 리셋 성공/실패 여부를 알린다. 소프트웨어 업데이트

트가 실패한 노드에 대해서는 사용자의 요구에 따라서 재시도 할 수 있도록 인터페이스를 제공한다.



(그림 4) 제안기법의 시퀀스 다이어그램

3. 구현 및 성능평가

3.1 구현 및 성능평가 환경

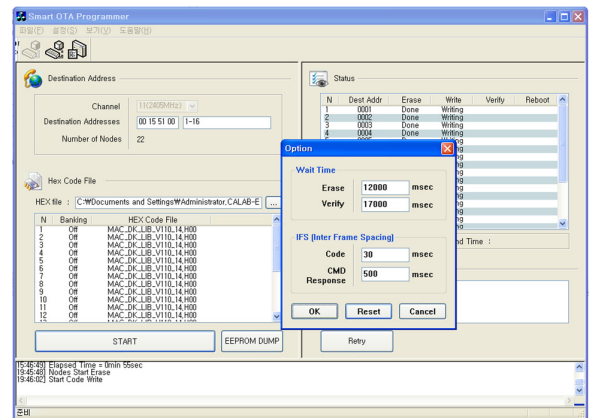
본 논문에서는 8051 MCU와 IEEE 802.15.4 표준을 따르는 RFIC, 64KB 외부 EEPROM으로 구성된 센서노드상에 제안 기법을 구현하였다. 통신을 위한 MAC 프로토콜은 IEEE 802.15.4 표준 중 데이터 송수신 기능만 동작 가능한 수준으로 구현된 라이브러리를 이용했다. 프로그래머로부터 센서노드로 전송되는 이미지는 C언어로 작성 및 컴파일 후 만들어지는 HEX파일(약 36KB)이며, 이미지 내에는 8051 MCU를 위한 기본 펌웨어, IEEE 802.15.4 MAC, 소프트웨어 업데이트 프로토콜이 포함되어 있으며, 센서노드에 부착된 7세그먼트를 동작시키는 응용 프로그램이 부가적으로 포함되어 있다.

3.2 PC GUI 응용 프로그램 구현

사용자가 프로그래머를 통한 이미지 전송을 편리하게 하기 위해서 프로그래머 장착된 PC(그림 2 참조)에서 동작하는 PC GUI응용 프로그램을 개발하였다. PC GUI 응용 프로그램은 프로그래머와 UART를 연결 한 뒤에 EEPROM삭제 확인 및 리셋 확인을 위해 업데이트 할 노드의 주소를 지정하고, 전송할 이미지(HEX)파일을 선택하면 인텔 HEX파일(intelhex)형식을 파싱해 로드할 주소별로 정렬한 후 64 바이트 씩 끊어 프로그래머로 UART를 통해 전송한다. 그림 5는 센서노드로 명령 및 소프트웨어 이미지를 전송하는 프로그래머를 동작시키기 위해 자체적으로 개발한 PC GUI 응용 프로그램의 동작 화면 이다. EEPROM삭제 대기 시간 및 소프트웨어 이미지를 전송할 때 패킷과 패킷 사이의 휴먼 시간, 리셋 확인 대기 시간 등을 옵션 창을 통해 지정 할 수 있도록 구현하였다.

3.3 소프트웨어 업데이트 기법 구현

제안된 기법은 센서노드 상에 포함된 IEEE 802.15.4 MAC을 이용하는 상위 네트워크 프로토콜 형태로 구현되었다. 그림 4에 나타난 소프트웨어 업데이트를 위한 각 명령 및 반환되는 데이터는 MAC 상의 데이터 패킷형태로 구현하였다. 각 센서노드의 주소(id)는 정적으로 할당하였으며, 모든 센서노드로의 브로드캐스트 전송을 위해서는 IEEE 802.15.4 표준에서 정한 주소를 이용하였다. 각 센서노드로 업데이트가 되는 이미지의 종류는 동일하며, 이미지가 수신될 때에 각 센서노드는 기억해 두었던 자신의 주소를 오버라이트 한다. 이때 새롭게 업데이트 될 이미지는 8051 MCU에서 지원하는 4개의 बैं크 중에서 3개 बैं크 크기를 넘지 않는다. 각 센서노드에서는 프로그래머로부터 수신하여 EEPROM에 저장해 둔 이미지를 8051 MCU 내의 프로그램 메모리로 카피할 때에 수행되는 프로그램은 MCU 내에 4개의 बैं크 중에서 이미지 업데이트가 발생하지 않는 마지막 बैं크를 이용한다.



(그림 5) MFC로 개발한 PC GUI 응용 프로그램

3.4 성능평가 결과

아래에서는 제안된 기법을 이용하여 다수의 센서노드를 업데이트 할 때 소요되는 시간을 측정하여본다. 우선, 한 센서노드를 업데이트 하는데 걸리는 시간을 각 단계별로 분석해 보았다. (표3 참조) 데이터 전송의 신뢰성을 확보하기 위해서 이미지 패킷 전송 간의 30msec의 휴먼 시간을 두었다. 반복 실험을 수행할 결과로 36KB 이미지를 전송하고 리셋 하는데 평균 51초가 소요되었다.

<표 3> 단일 센서노드 업데이트 시 단계별 소요시간 측정결과

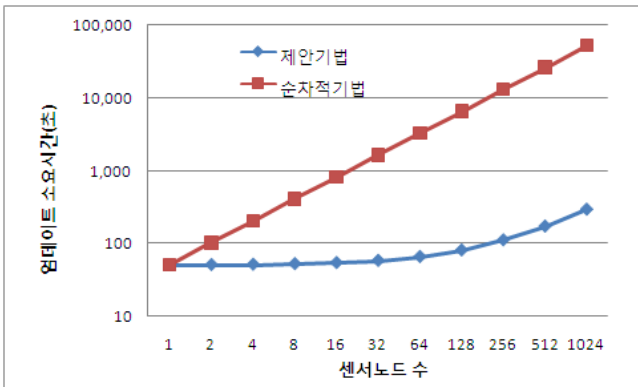
단 계	평균소요시간
EEPROM 삭제 및 확인	11.5초
이미지 전송 & EEPROM 쓰기	23.5초
체크섬 계산 및 리셋 시간	16.0초
합 계	51.0초

다음으로, 센서노드 수를 늘여서 22개 노드를 대상으로 동일한 실험을 수행하였다 (표4 참조) 단일 센서노드 업데이트에 비해서 EEPROM 삭제확인 및 리셋확인 명령을 각 노드별로 수행할 때 센서노드 하나당 평균 120m초 정도의 부가적인 오버헤드가 존재함을 볼 수 있었다. 48개의 센서노드를 대상으로 실험했을 경우도 비슷한 오버헤드가 존재함을 확인할 수 있었다.

<표 4> 22개 센서노드 업데이트시 단계별 소요시간 측정결과

단 계	평균소요시간
EEPROM 삭제 및 확인	14.5초
이미지 전송 & EEPROM 쓰기	23.5초
체크섬 계산 및 리셋 시간	18.0초
합 계	56.0초

끝으로, 그림 6에 센서노드가 증가하면서 소요되는 업데이트 시간을 표시하였다. 제안 기법은 센서노드들 간의 병렬성을 극대화 함으로써 본 기법을 적용하지 않은 순차적 소프트웨어 업데이트에 비해 업데이트 시간이 현저하게 소요됨을 볼 수 있다.



(그림 6) 센서노드 수에 따른 업데이트 총 소요시간 비교

실험결과 EEPROM 삭제 확인, 리셋 확인 과정에서 1:1 통신을 하기 때문에 노드 수가 늘어남에 따라 전체 시간이 늘어나긴 하지만 약 50개의 노드를 1분 정도의 시간으로 업데이트가 가능하므로 일일이 노드마다 새로운 소프트웨어를 프로그래밍 하는 시간에 비하면 충분히 효율적이라고 할 수 있다. 단, 무선 통신이므로 주위에 다른 센서 네트워크가 동작하는 등의 방해요인이 있을 경우 업데이트에 실패하는 노드가 많아짐을 볼 수 있었다.

4. 결론 및 향후계획

본 논문에서는 8051 MCU와 IEEE 802.15.4 표준을 따르는 RFIC, 64KB 외부 EEPROM으로 구성된 센서노드를 이용해 병렬성을 극대화한 무선 소프트웨어 업데이트 기법을 제안 및 구현하고 그 성능을 평가했다. 성능 평가 결과에 따르면 대량의 센서노드가 설치되는 네트워크에서 짧은 시간에 많은 수의 노드를 업데이트 할 수 있어 무선 센서네트워크를 연구 및 개발단계에서 유용 할 것으로 예상할 뿐 아니라, 조밀하게 설치된 제품에도 하드웨어적인 특성만 맞으면 초기 소프트웨어 업데이트 프로토콜 설치 과정을 거쳐 사용할 수 있을 것으로 기대한다.

향후 연구로 제안 기법을 기반으로 향후 멀티 홉 네트워크 환경에 적용할 수 있는 기법을 연구하고, 업데이트할 소프트웨어 이미지 전체가 아닌 변경되는 일부 이미지만 변경, 전송하여 전체노드의 업데이트 시간을 줄일 수 있는 기법을 개발할 예정이다.

참고문헌

- [1] Q. Wang, Y. Zhu and L. Cheng, "Reprogramming Wireless Sensor Networks: Challenges and Approaches," IEEE Network, Vol. 20, No. 3, pp. 48-55, May/June. 2006.
- [2] S. Brown, "Updating Software in Wireless Sensor Networks: A Survey," Technical Report UCC-CS-2006-13-07.
- [3] M. Kuorilehto, M. Hannikainen and T. Hamalainen, "A Survey of Application Distribution in Wireless Sensor Networks," EURASIP Journal on Wireless Communications and Networking, 2005.
- [4] N. Reijers and K. Langendoen, "Efficient Code Distribution in Wireless Sensor Networks," Proc. 2nd ACM Int'l. Conf. Wireless Sensor Networks and Apps., pp. 60 - 67, 2003.
- [5] J. Jeong and D. Culler, "Incremental Network Programming for Wireless Sensors," Proc. 1st Annual IEEE ComSoc. Conf. Sensor and Ad Hoc Commun. and Networks (SECON), pp. 25-33, 2004.
- [6] P. Marrón, A. Lachenmann, D. Minder, J. Hähner, R. Sauter and K. Rothenmel, "TinyCubus: A flexible and adaptive framework for sensor networks," EWSN, 2005.
- [7] P. Levis, N. Patel, D. Culler and S. Shenker, "Trickle: A Self-Regulating Algorithm for Code Propagation and Maintenance in Wireless Sensor Networks," Proc. USENIX/ACM Symposium on NSDI, 2004.
- [8] Crossbow Technology, Inc., "Mote In-Network Programming User Reference" and "Mica2 Wireless Measurement System Datasheet," 2003.
- [9] S. Kulkarni and L. Wang, "MNP: Multihop Network Programming for Sensor Networks," International Conference on Distributed Computing Systems, Columbus OH, Jun. 2005.
- [10] J. Hui and D. Culler. "The Dynamic Behavior of a Data Dissemination Protocol for Network Programming at Scale," Proc. ACM SenSys, pp. 81 - 94, 2004.
- [11] T. Stathopoulos, J. Heidemann, and D. Estrin, "A Remote Code Update Mechanism for Wireless Sensor Networks," Center for Embedded Network Sensing. Technical Reports. Paper 37. Jan. 1, 2003.
- [12] S. Kulkarni and M. Arumugam, "Infuse: A TDMA Based Data Dissemination Protocol for Sensor Networks," Conference on Embedded Networked Sensor Systems 2004.
- [13] V. Naik, A. Arora, P. Sinha and H. Zhang, "Sprinkler: A Reliable and Energy Efficient Data Dissemination Service for Extreme Scale Wireless Networks of Embedded Devices," IEEE Transactions on Mobile Computing, 6(7):762-776, Jul. 2007
- [14] L. Phillips, "Aqueduct: Robust and Efficient Code Propagation in Heterogeneous Wireless Sensor Networks," Master's thesis, Univ. Colorado, 2005.
- [15] <http://www.keil.com/support/man/docs/lx51/>