



인하공업전문대학
INHA TECHNICAL COLLEGE

무선 센서 네트워크 4주차

인하공업전문대학 컴퓨터 정보과
김한결

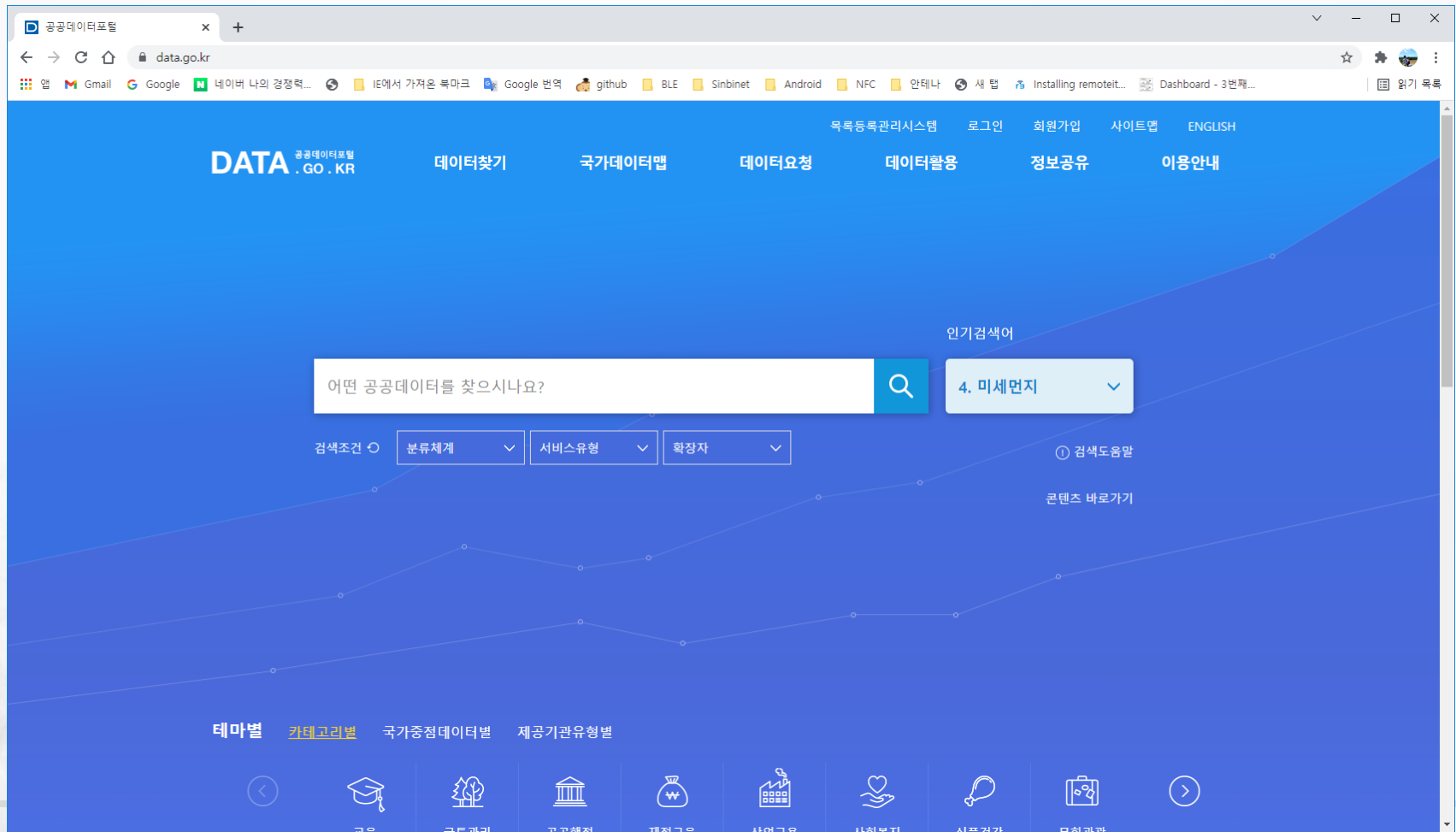
- 요약

- I. 공공데이터 API 사용 (미세먼지 데이터 수집)
- II. TinyOS 환경구축
- III. TinyOS 온습도 모니터링

공공데이터 API - 미세먼지

❖ 공공데이터 포털 URL

: <https://www.data.go.kr>



❖ 회원가입 : 이후 절차 생략

STEP1 가입확인 및 회원구분

STEP2 정보입력 및 약관동의

1. 회원가입여부 확인을 위해 이름, 이메일 주소를 입력해주세요.

이름

※ 실명으로 가입하지 않을 경우 공공데이터 제공신청, 분쟁조정신청 등 일부 서비스에 대해 불이익이 있을 수 있습니다.

이메일

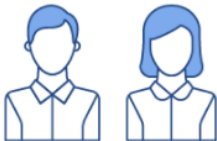
sample@sample.com

가입확인

2. 14세이상은 일반회원으로 가입해주세요

일반인

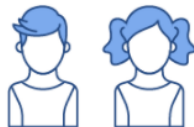
만14세이상
내국인



가입하기

어린이

만14세미만
내국인



가입하기

공공데이터 API

❖ 미세먼지 검색

The screenshot shows the homepage of the data.go.kr portal. The browser's address bar displays 'data.go.kr/index.do'. The page features a blue header with the 'DATA GO . KR' logo and navigation links: '데이터찾기', '국가데이터맵', '데이터요청', '데이터활용', '정보공유', and '이용안내'. A top navigation bar includes links for '목록등록관리시스템', '로그인', '회원가입', '사이트맵', and 'ENGLISH'. A central search area is highlighted with a red box, containing a search input field with '미세먼지' entered, a magnifying glass icon, and a dropdown menu showing '4. 미세먼지'. Below the search bar, there are links for '검색도움말' and '콘텐츠 바로가기'. At the bottom, a '테마별' (By Theme) section lists various categories: '카테고리별', '국가중점데이터별', and '제공기관유형별'. The categories are represented by icons and text: '교육' (Education), '국토관리' (Land Management), '공공행정' (Public Administration), '재정금융' (Finance), '산업고용' (Industry and Employment), '사회복지' (Social Welfare), '식품건강' (Food and Health), and '문화관광' (Culture and Tourism).

❖ 미세먼지 검색

: -> 오픈API

데이터목록

미세먼지



연관 황산화물 초미세먼지 대기오염물질 먼지 저감 환경오염 대기환경 황사 호흡기질환 경유차

제공기관별 검색

상세검색

"미세먼지"에 대해 총 183건이 검색되었습니다.

조건검색

초기화

분류체계	서비스유형	제공기관유형	태그	확장자
------	-------	--------	----	-----

국가중점데이터 분류 조건 추가하기 +

조건열기

전체(183건)	파일데이터(145건)	오픈 API(38건)	표준데이터셋0개(0건)
----------	-------------	-------------	--------------

정확도순

5개씩

❖ 미세먼지 검색

: -> 오픈API -> 한국환경공단_에어코리아_대기오염정보 -> 활용신청

환경기상

공공기관

[미리보기](#)

XML

JSON

한국환경공단_에어코리아_대기오염정보

각 측정소별 대기오염정보를 조회하기 위한 서비스로 기간별, 시도별 대기오염 정보와 통합대기환경지수 나쁨 이상 측정소 내역, 대기질(미세먼지/오존) 예보 통보 내역 등을 조회할 수...

제공기관 한국환경공단 수정일 2021-11-11 조회수 36855 활용신청 7499 키워드 한국환경공단,에어코리아,미세먼지

활용신청

❖ 활용신청

: -> 오픈API -> 한국환경공단_에어코리아_대기오염정보 -> 활용신청

: 활용목적 -> 기타(교육용 테스트)

공공데이터 제공제도

- * 공공데이터중 위치정보를 포함한 서비스를 사용하고자 하는 사업자는 '위치정보의 보호 및 이용 등에 관한 법률'에 따라 방송통신위원회에 '위치정보서비스 허가'를 받거나 '위치기반 서비스사업 신고'를 하여야 합니다.
- * 이에 해당하는 사업자인 경우에는 첨부파일에 '위치기반서비스사업신고필증'을 첨부해 주시기 바랍니다.
- * 활용신청 시 '위치기반서비스사업신고필증'이 등록되지 않으면 반려가 될 수 있으니 참고 하시기 바랍니다.

활용목적 선택

*표시는 필수 입력항목입니다.

*활용목적

☐ 웹 사이트 개발 ☐ 앱개발 (모바일 솔루션등) ☒ 기타 ☐ 참고자료 ☐ 연구(논문 등)

교육용 테스트

7/250

첨부파일

파일 선택

Drag & Drop으로 파일을 선택 가능합니다.

❖ 활용신청

: 시도별 실시간 측정정보 조회 체크 -> 동의합니다 -> 활용신청

상세기능정보 선택

<input type="checkbox"/>	상세기능	설명	일일 트래픽
<input type="checkbox"/>	대기질 예보통보 조회	통보코드와 통보시간으로 예보정보와 발생 원인 정보를 조회하는 대기질(미세먼지/오존) 예보통보 조회	500
<input type="checkbox"/>	초미세먼지 주간예보 조회	통보코드와 통보시간으로 대기질 전망과 주간예보 정보를 조회하는 초미세먼지 주간예보통보 조회	500
<input checked="" type="checkbox"/>	측정소별 실시간 측정정보 조회	측정소명과 측정데이터 기간(일,한달,3개월)으로 해당 측정소의 일반항목 측정정보를 제공하는 측정소별 실시간 측정정보 조회	500
<input type="checkbox"/>	통합대기환경지수 나쁨 이상 측정소 목록조회	통합대기환경지수가 나쁨 등급 이상인 측정소명과 주소 목록 정보를 제공하는 통합대기환경지수 나쁨 이상 측정소 목록조회	500
<input checked="" type="checkbox"/>	시도별 실시간 측정정보 조회	시도명을 검색조건으로 하여 시도별 측정소목록에 대한 일반항목과 CAI최종 실시간 측정값과 지수 정보 조회 기능을 제공하는 시도별 실시간 측정정보 조회	500

라이선스 표시

*이용허락범위

저작자표시-변경금지

(사유 : 출처 및 데이터 오류가능성 표시)

☒ 동의합니다.

취소

활용신청

❖ 개발계정

상세보기화면

1. API 문서 다운

2. 미리보기 확인

데이터 활용 >

데이터 요청 >

나의 문의 >

회원정보 수정 >

기본정보

데이터명	한국환경공단_에어코리아_대기오염정보 상세설명		
서비스유형	REST	심의여부	자동승인
신청유형	개발계정 연정신청	처리상태	승인
활용기간	2021-11-12 ~ 2026-04-08		

서비스정보

참고문서	한국환경공단_에어코리아_OpenAPI 기술문서.zip <- API 문서
데이터포맷	JSON+XML
End Point	https://apis.data.go.kr/B552584/ArpltnInforInquireSvc

API 환경 또는 API 호출 조건에 따라 인증키가 적용되는 방식이 다를 수 있습니다.
포털에서 제공되는 Encoding/Decoding 된 인증키를 적용하면서 구동되는 키를 사용하시기 바랍니다.
* 향후 포털에서 더 명확한 정보를 제공하기 위해 노력하겠습니다.

일반 인증키 (Encoding)	
일반 인증키 (Decoding)	

활용신청 상세기능정보

NO	함	
1	측정소별 실시간 측정정보 조회	해당 측정소의 일반항목 측정정보를 제공하는 측정소별 실시간 측정정보조회

미리보기

확인

❖ 활용신청 상세기능정보

활용신청 상세기능정보

NO	상세기능	설명	일일 트래픽	미리보기
1	측정소별 실시간 측정정보 조회	측정소명과 측정데이터 기간(일,한달,3개월)으로 해당 측정소의 일반항목 측정정보를 제공하는 측정소별 실시간 측정정보조회	500	확인

요청변수(Request Parameter)

[닫기](#)

항목명	샘플데이터	설명
serviceKey	인증키(URL Encode)	공공데이터포털에서 받은 인증키
returnType	xml	xml 또는 json
numOfRows	100	한 페이지 결과 수
pageNo	1	페이지번호
stationName	종로구	측정소 이름
dataTerm	DAILY	요청 데이터기간(1일: DAILY, 1개월: MONTH, 3개월: 3MONTH)
ver	1.0	버전별 상세 결과 참고

[미리보기](#)

❖ 활용신청 상세기능정보 : 미리보기

CYbYJHlswWquBvkKR0PKmUHppucbBJztsGx17tq9cx56sDqItq3Ltd8wIqay9PP6yoyUmDQ

```
<response>
  <header>
    <resultCode>00</resultCode>
    <resultMsg>NORMAL_CODE</resultMsg>
  </header>
  <body>
    <items>
      <item>
        <so2Grade>1</so2Grade>
        <coFlag/>
        <khaiValue>42</khaiValue>
        <so2Value>0.003</so2Value>
        <coValue>0.3</coValue>
        <pm25Flag/>
        <pm10Flag/>
        <pm10Value>30</pm10Value>
        <o3Grade>1</o3Grade>
        <khaiGrade>1</khaiGrade>
        <pm25Value>10</pm25Value>
        <no2Flag/>
        <no2Grade>1</no2Grade>
        <o3Flag/>
        <pm25Grade>1</pm25Grade>
        <so2Flag/>
        <dateTime>2021-11-12 12:00</dateTime>
        <coGrade>1</coGrade>
        <no2Value>0.010</no2Value>
        <pm10Grade>1</pm10Grade>
        <o3Value>0.025</o3Value>
      </item>
      <item>
        <so2Grade>1</so2Grade>
        <coFlag/>
        <khaiValue>33</khaiValue>
        <so2Value>0.003</so2Value>
        <coValue>0.4</coValue>
        <pm25Flag/>
        <pm10Flag/>
        <pm10Value>28</pm10Value>
        <o3Grade>1</o3Grade>
        <khaiGrade>1</khaiGrade>
        <pm25Value>9</pm25Value>
        <no2Flag/>
        <no2Grade>1</no2Grade>
        <o3Flag/>
        <pm25Grade>1</pm25Grade>
        <so2Flag/>
        <dateTime>2021-11-12 11:00</dateTime>
        <coGrade>1</coGrade>
        <no2Value>0.015</no2Value>
        <pm10Grade>1</pm10Grade>
        <o3Value>0.020</o3Value>
      </item>
    </items>
  </body>
</response>
```

ionName=주안&dataTerm=DAILY&ver=

❖ 미세먼지 OpenAPI 문서

다. 상세기능내역

1) 측정소별 실시간 측정정보 조회 상세기능명세

a) 상세기능정보

상세기능 번호	1	상세기능 유형	조회(목록)
상세기능명(국문)	측정소별 실시간 측정정보 조회		
상세기능 설명	측정소명과 측정데이터 기간(일,한달,3 개월)으로 해당 측정소의 일반항목 측정정보를 제공하는 측정소별 실시간 측정정보조회		
Call Back URL	http://apis.data.go.kr/B552584/ArpltnInforInquireSvc/getMsrstnAcctoRltmMesureDnsty		
최대 메시지 사이즈	[1000] byte		
평균 응답 시간	[500] ms	초당 최대 트래잭션	[50] tps

❖ 미세먼지 OpenAPI 문서

b) 요청 메시지 명세

항목명(영문)	항목명(국문)	항목크기	항목구분	샘플데이터	항목설명
serviceKey	서비스키	-	1	인증키(URL Encode)	서비스키
returnType	데이터표출방식	4	0	xml	데이터 표출방식 xml 또는 json
numOfRows	한 페이지 결과 수	4	0	100	한 페이지 결과 수
pageNo	페이지 번호	4	0	1	페이지 번호
stationName	측정소명	30	1	종로구	측정소 이름
dataTerm	데이터기간	10	1	DAILY	요청 데이터기간(1 일: DAILY, 1 개월: MONTH, 3 개월: 3MONTH)
ver	오퍼레이션 버전	4	0	1.0	버전별 상세 결과 아래쪽 참고

※ 항목구분 : 필수(1), 옵션(0)

※ 버전(ver) 항목설명

- 버전을 포함하지 않고 호출할 경우 : PM_{2.5} 데이터가 포함되지 않은 원래 오퍼레이션 결과 표출.
- 버전 1.0 을 호출할 경우 : PM_{2.5} 데이터가 포함된 결과 표출.
- 버전 1.1 을 호출할 경우 : PM₁₀, PM_{2.5} 24 시간 예측이동 평균데이터가 포함된 결과 표출.
- 버전 1.2 을 호출할 경우 : 측정망 정보 데이터가 포함된 결과 표출.
- 버전 1.3 을 호출할 경우 : PM₁₀, PM_{2.5} 1 시간 등급 자료가 포함된 결과 표출.

❖ Python3 구현

: dustAPI.py

```
1 from bs4 import BeautifulSoup
2 from urllib.request import Request, urlopen
3 from urllib.parse import urlencode, quote_plus, unquote
4 import requests
5
6 url = 'http://apis.data.go.kr/B552584/ArpltnInforInquireSvc/getMsrstnAcctoRltmMesureDnsty'
7 queryParams = '?' + urlencode({quote_plus('serviceKey') :
8 'Gkhq+...&MqubVhKkR0PkmUhpucbBjzteGx17tg0ex56eDqItq3L...&Q=='
9                                     ,quote_plus('returnType'):'xml'
10                                    ,quote_plus('numOfRows'):'10'
11                                    ,quote_plus('pageNo'):'1'
12                                    ,quote_plus('stationName'):'주안'
13                                    ,quote_plus('dataTerm'):'DAILY'
14                                    ,quote_plus('ver'):'1.0'})
15
16 res = requests.get(url+queryParams)
17 soup = BeautifulSoup(res.content, 'html.parser')
18 data = soup.find_all('item')
19 print(data)
20
21 for item in data:
22     dataterm = item.find('datetime')
23     pm25value = item.find('pm25value')
24     print(pm25value.get_text())
25     print(dataterm.get_text())
26
```

Python3 미세먼지 API 구현

❖ Python3 구현

: bs4, requests라이브러리 설치

```
RESTART: C:/Users/PC/AppData/Local/Programs/Python/Python313/Python313.exe
Traceback (most recent call last):
  File "C:/Users/PC/AppData/Local/Programs/Python/Python313/Python313.exe", line 1, in <module>
    from bs4 import BeautifulSoup
ModuleNotFoundError: No module named 'bs4'
>>>
```

```
RESTART: C:/Users/PC/AppData/Local/Programs/Python/Python313/Python313.exe
Traceback (most recent call last):
  File "C:/Users/PC/AppData/Local/Programs/Python/Python313/Python313.exe", line 1, in <module>
    import requests
ModuleNotFoundError: No module named 'requests'
>>>
```


❖ Python3 구현

: bs4, requests라이브러리 설치 - 1

```
명령 프롬프트
Microsoft Windows [Version 10.0.22631.5624]
(c) Microsoft Corporation. All rights reserved.

C:\Users\PC>cd C:\Users\PC\AppData\Local\Programs\Python\Python313\Scripts

C:\Users\PC\AppData\Local\Programs\Python\Python313\Scripts>dir/w
C 드라이브의 볼륨에는 이름이 없습니다.
볼륨 일련 번호: 56CB-E89D

C:\Users\PC\AppData\Local\Programs\Python\Python313\Scripts 디렉터리

[.]                [..]                pip.exe            pip3.13.exe        pip3.exe
                  3개 파일                325,176 바이트
                  2개 디렉터리  159,749,652,480 바이트 남음

C:\Users\PC\AppData\Local\Programs\Python\Python313\Scripts>pip3.13.exe install 라이브러리이름|
```

bs4, requests

- K-Mote

제품 구성

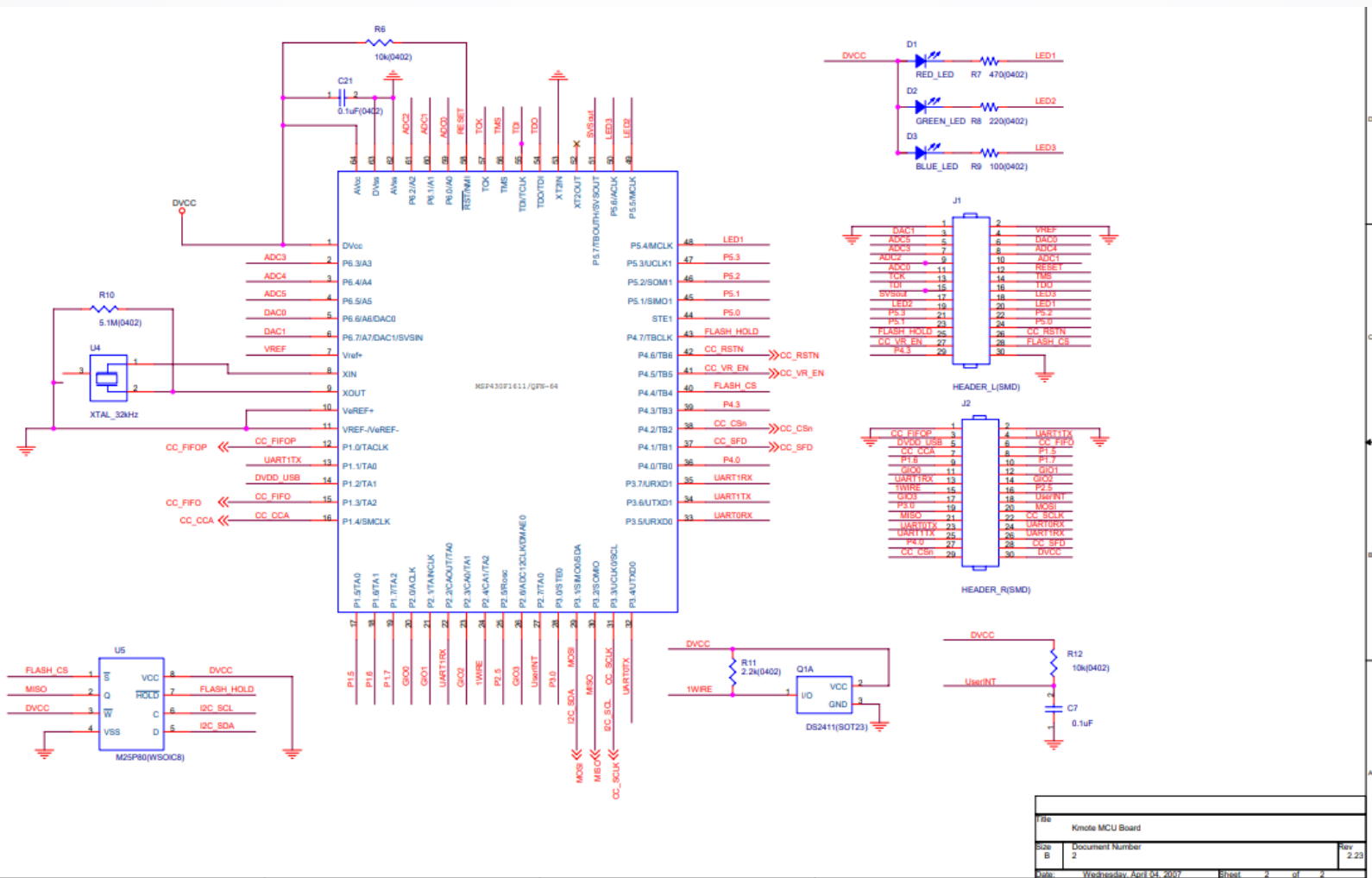
- Radio 모듈 (윗면)
- Uploading 모듈(아랫면)
- Kmot-B Datasheet



제품 특징

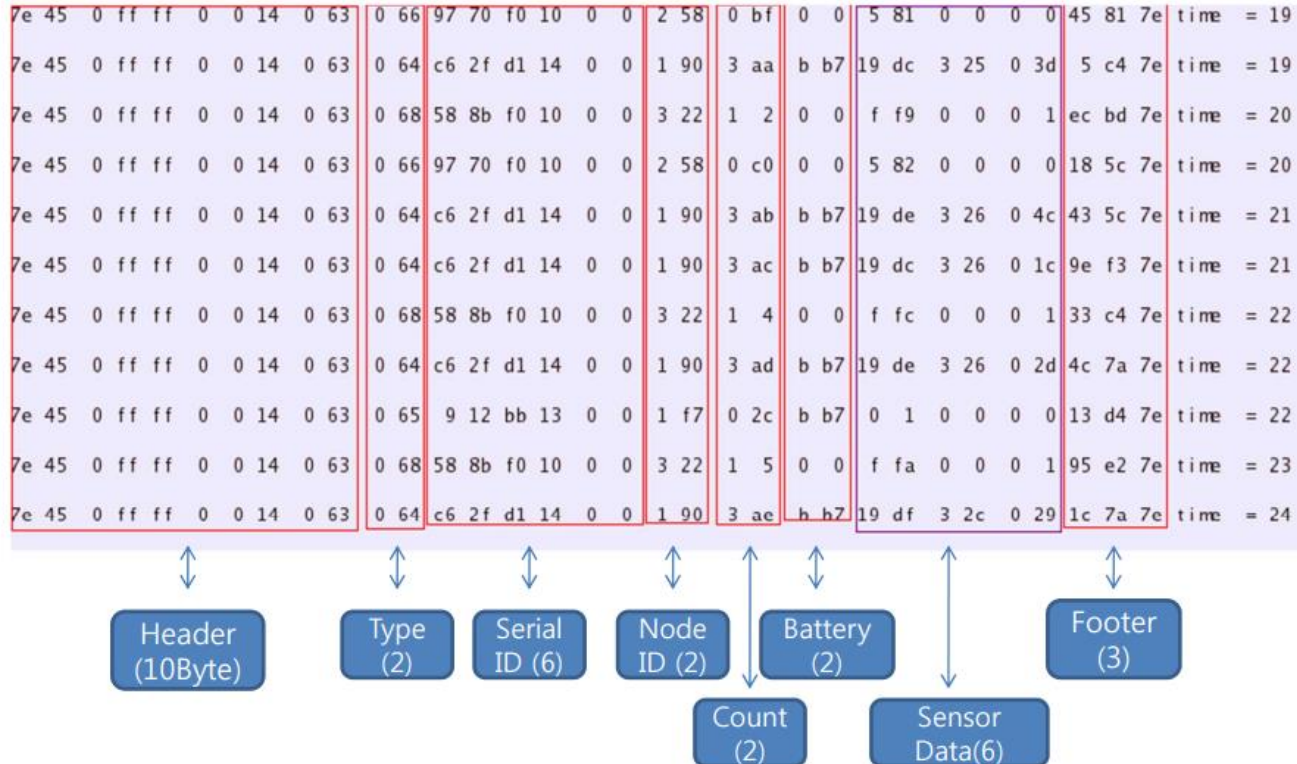
- TelosB 호환 제품
- USB 커넥터로 TinyOS 이미지 업로딩
- 8MHz Texas Instrument사의 MSP430F1611 마이크로 프로세서 탑재
- 250 kbps 데이터 전송률
- 2.4 GHz 통신 주파수 허용
- IEEE 802.15.4 무선 Radio 칩(CC2420) 탑재
- 60 pin 커넥터
- 세라믹 안테나 사용(실내 30M 통신 가능)
- 제품 크기 : 40 mm X 22 mm X 8 mm

- K-Mote 회로도



- TinyOS NesC Packet Protocol

Sensor Data Packet



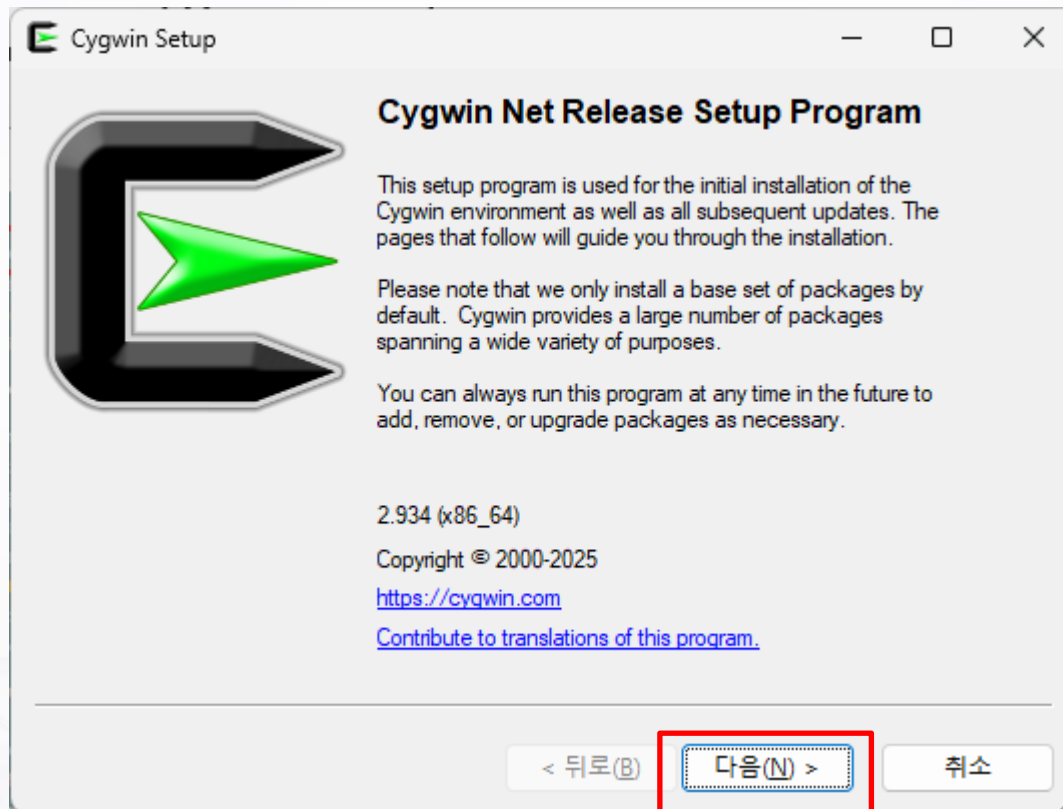
Windows 기반 TinyOS 환경 구축

- Cygwin 설치
- <https://www.cygwin.com/install.html>

The screenshot shows the Cygwin website with the URL [cygwin.com/install.html](https://www.cygwin.com/install.html) in the browser address bar. The left sidebar contains a navigation menu with the following items: Cygwin, Install Cygwin (highlighted with a red box), Update Cygwin, Search Packages, Licensing Terms, Cygwin/X, Community, Reporting Problems, Mailing Lists, Newsgroups, IRC channels, Gold Stars, Mirror Sites, Donations, Documentation, FAQ, User's Guide, API Reference, Acronyms, Contributing, Source in Git, Cygwin DLL, Cygwin Packages, Cygwin Apps, and Related Sites. The main content area features the Cygwin logo and the tagline "Get that [Linux](#) feeling - on Windows". Below this is the heading "Installing and Updating Cygwin Packages". A sub-heading "Installing and Updating Cygwin for 64-bit versions of Windows" is followed by the text "Run [setup-x86_64.exe](#) any time you want to update or install a Cygwin package for 64-bit windows." and "The gpg [signature](#) for [setup-x86_64.exe](#) can be used to verify the validity of this binary using the public key [here](#)." Another sub-heading "General installation notes" is followed by two paragraphs: "When installing packages for the first time, the setup program *does not install every package*. Only the **minimal base packages** from the Cygwin distribution are installed by default, which takes up about 100 MB." and "Clicking on categories and packages in the setup program package installation screen allows you to select what is installed or updated." A third paragraph states "Individual packages like *bash*, *gcc*, *less*, etc. are released independently of the Cygwin DLL, so the Cygwin DLL version is not useful as a general Cygwin release number. The setup program tracks the versions of all installed components and provides the mechanism for **installing** or **updating** everything available from this site for Cygwin." A fourth paragraph says "Once you've installed your desired subset of the Cygwin distribution, the setup program will remember what you selected, so re-running it will update your system with any new package releases." The final paragraph reads "The setup program will check by default if it runs with administrative privileges and, if not, will try to elevate the process. If you want to avoid this behaviour and install under an unprivileged account just for your own usage, run setup with the `--no-admin` option."

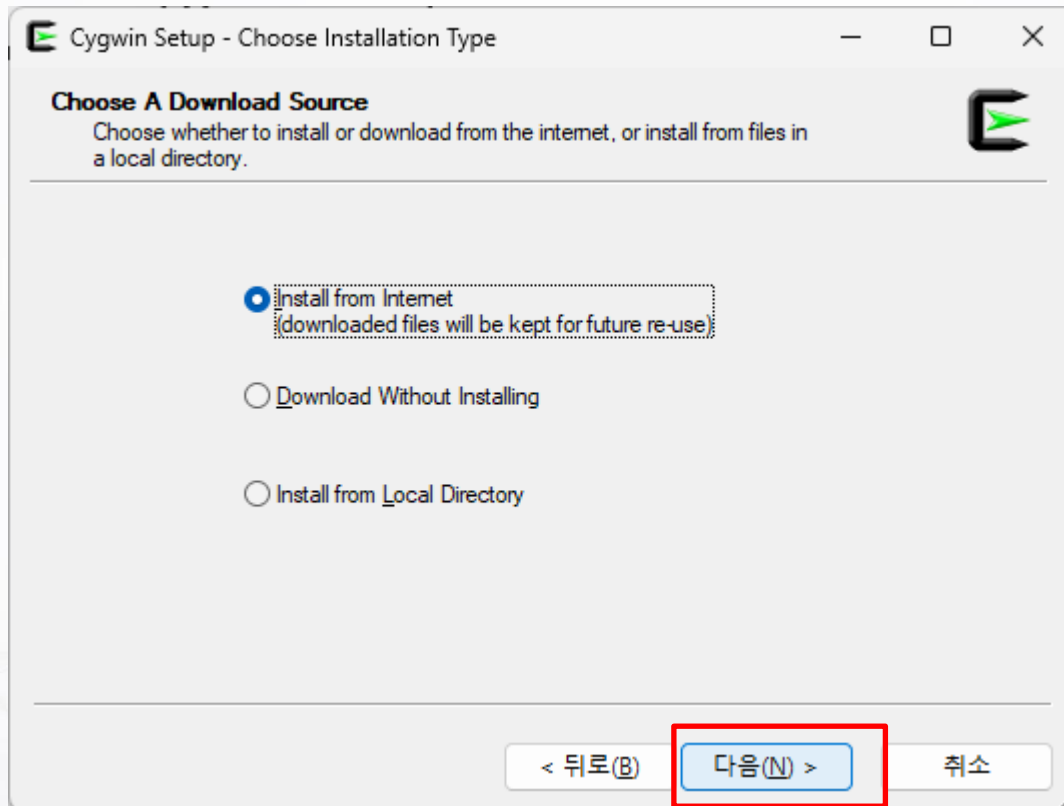
Windows 기반 TinyOS 환경 구축

- Cygwin 설치
- <https://www.cygwin.com/install.html>



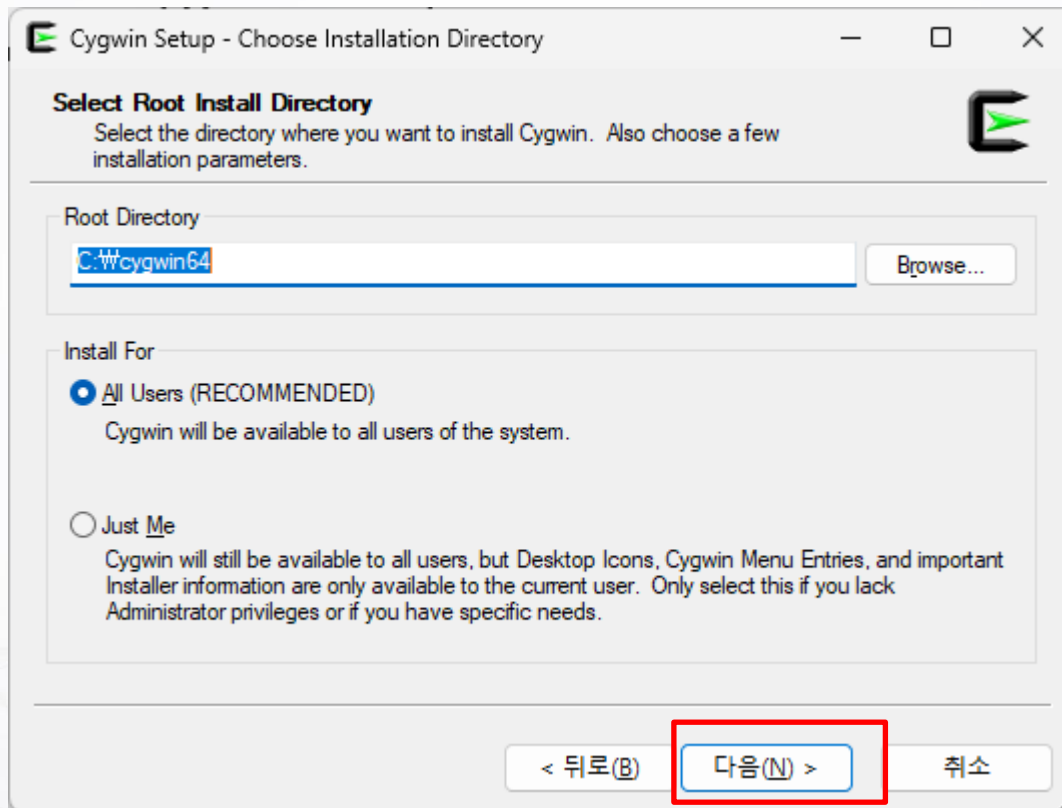
Windows 기반 TinyOS 환경 구축

- Cygwin 설치
- <https://www.cygwin.com/install.html>



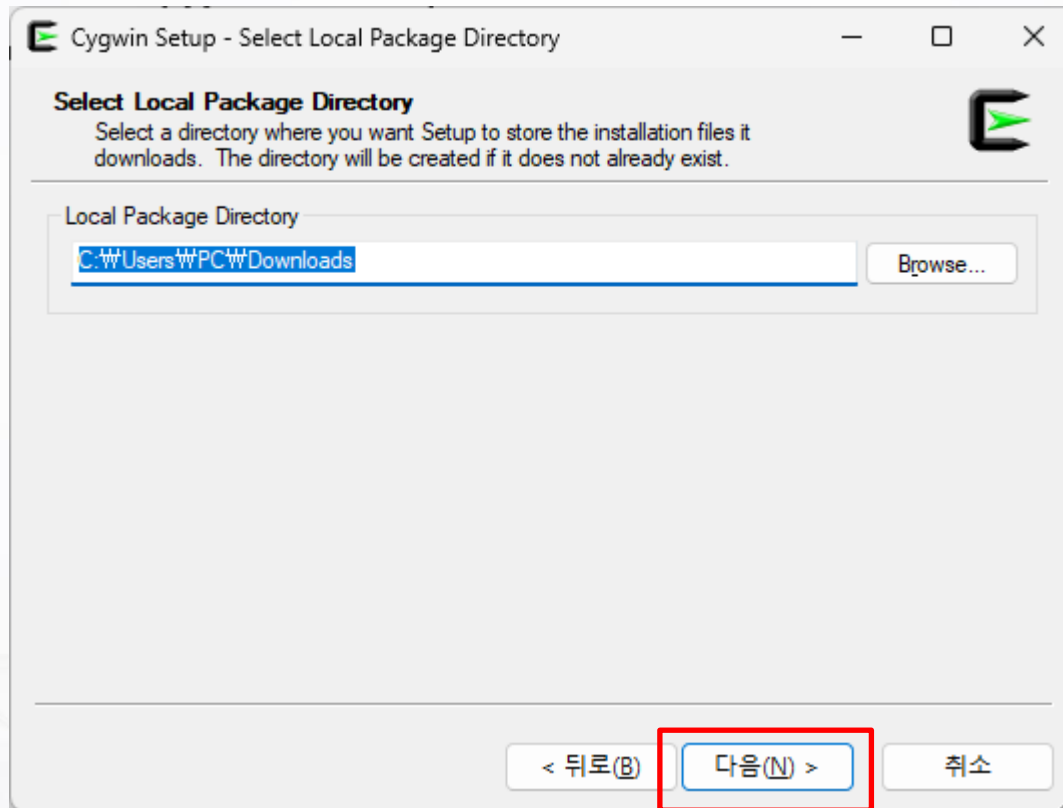
Windows 기반 TinyOS 환경 구축

- Cygwin 설치
- <https://www.cygwin.com/install.html>



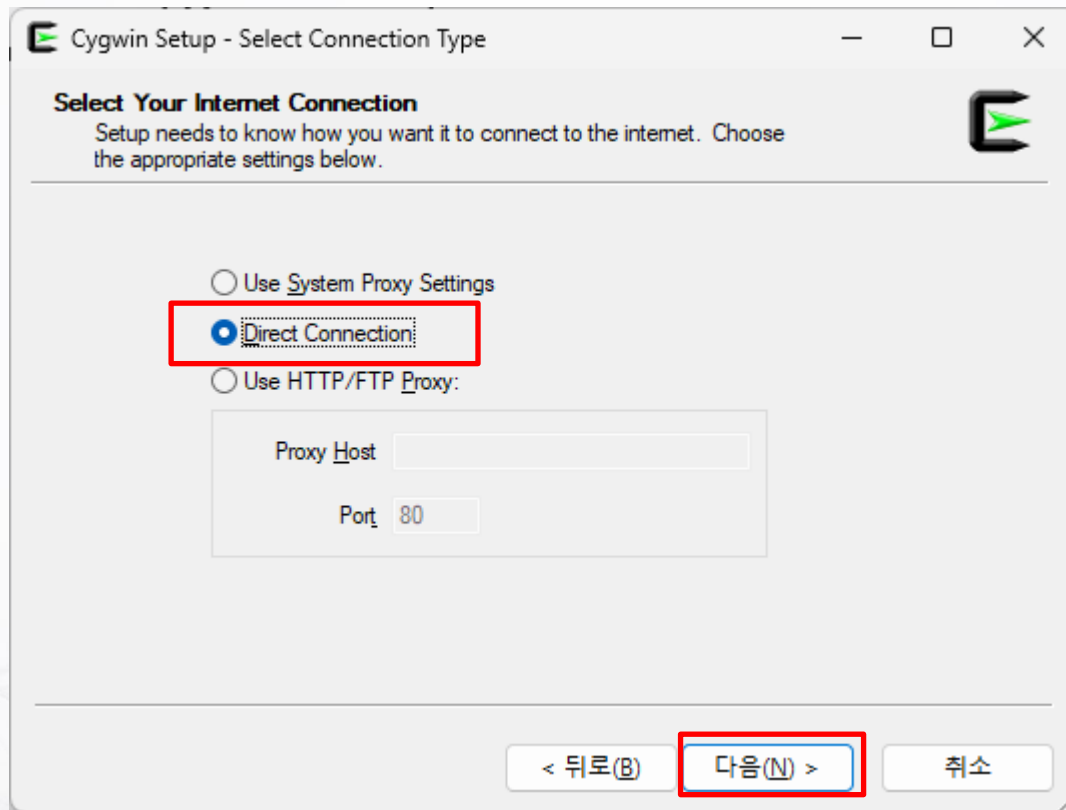
Windows 기반 TinyOS 환경 구축

- Cygwin 설치
- <https://www.cygwin.com/install.html>



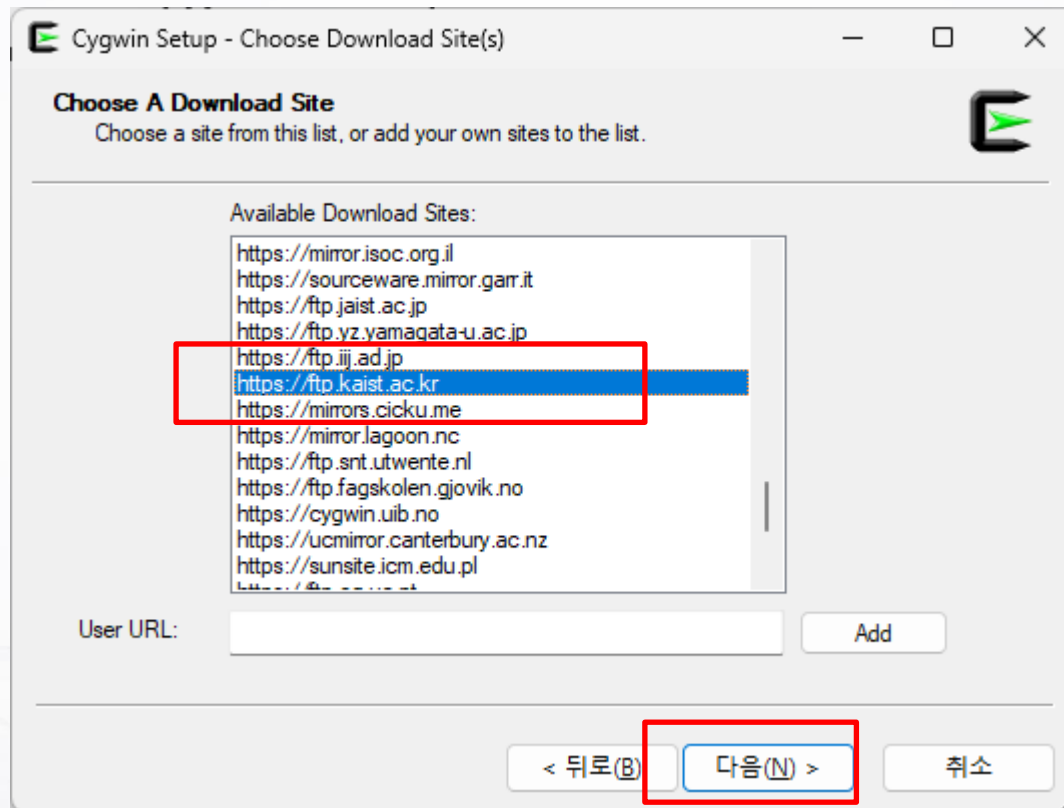
Windows 기반 TinyOS 환경 구축

- Cygwin 설치
- <https://www.cygwin.com/install.html>



Windows 기반 TinyOS 환경 구축

- Cygwin 설치
- <https://www.cygwin.com/install.html>



Windows 기반 TinyOS 환경 구축

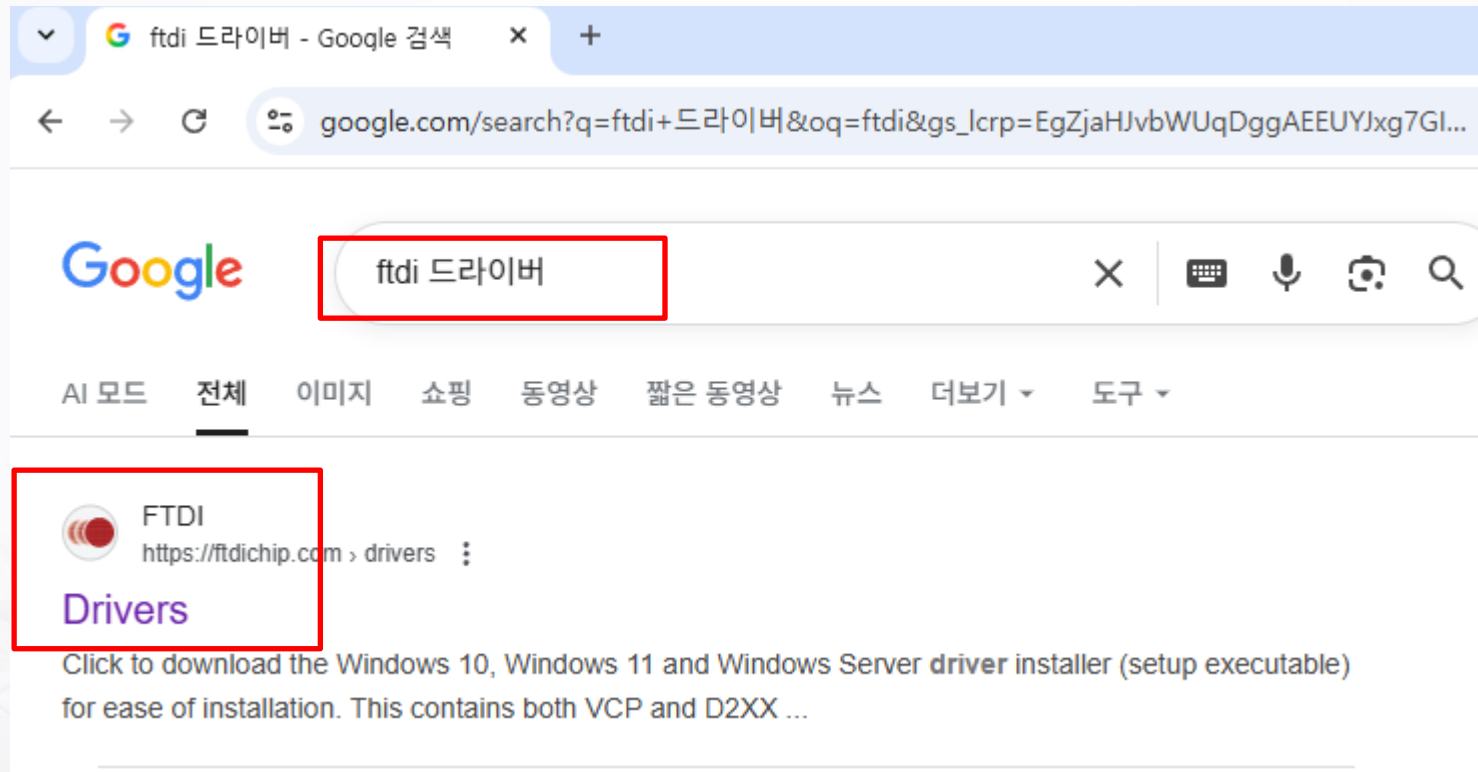
- Cygwin K-mote(TinyOS) 환경구축
- FTDI 드라이버 설치

```
/opt/tinyos-2.x/apps/Blink
mkdir -p build/telosb
  compiling BlinkAppC to a telosb binary
ncc -o build/telosb/main.exe -Os -fnesc-separator=__ -Wall -Wshadow -Wnesc-all
-target=telosb -fnesc-cfile=build/telosb/app.c -board= -DDEFINED_TOS_AM_GROUP=0x
22 -DIDENT_APPNAME=W"BlinkAppCW" -DIDENT_USERNAME=W"PCW" -DIDENT_HOSTNAME=W"4-40
5-██████W" -DIDENT_USERHASH=0x6f10f836L -DIDENT_TIMESTAMP=0x68cb5ce8L -DIDENT_UI
DHASH=0x557c215dL BlinkAppC.nc -lm
  compiled BlinkAppC to build/telosb/main.exe
      2538 bytes in ROM
      56 bytes in RAM
msp430-objcopy --output-target=ihex build/telosb/main.exe build/telosb/main.ihex
  writing TOS image
tos-set-symbols --objcopy msp430-objcopy --objdump msp430-objdump --target ihex
build/telosb/main.ihex build/telosb/main.ihex.out-10 TOS_NODE_ID=10 ActiveMessag
eAddressC__addr=10
Could not find symbol ActiveMessageAddressC__addr in build/telosb/main.exe, igno
ring symbol.
Could not find symbol TOS_NODE_ID in build/telosb/main.exe, ignoring symbol.
error, could not open key HKLM\SYSTEM\CurrentControlSet\Enum\WFTDIBUS
  found no motes (using bsl,auto)
make: *** [/opt/tinyos-2.x/support/make/msp/bsl.extra:42: bsltest] Error 1

PC@4-405-██████ /opt/tinyos-2.x/apps/Blink
$ |
```

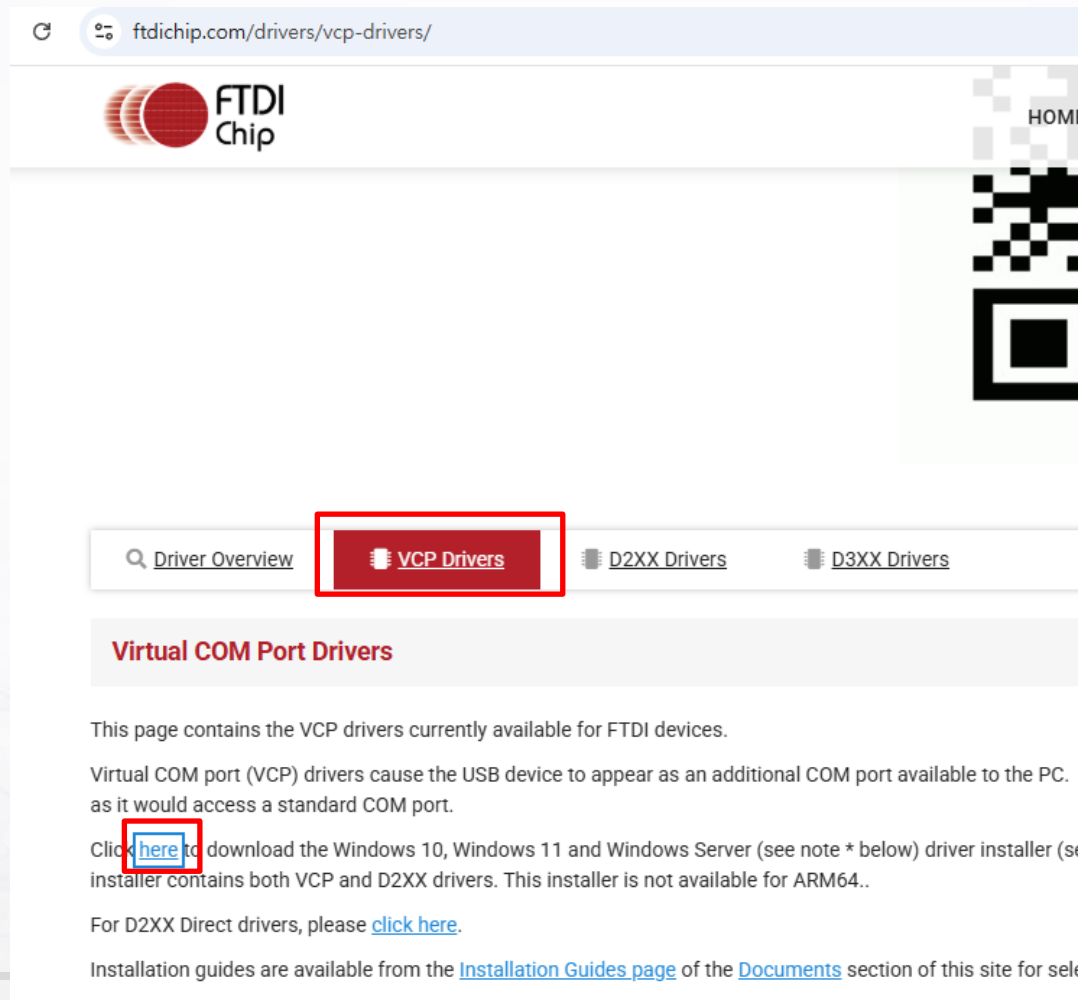
Windows 기반 TinyOS 환경 구축

- Cygwin K-mote(TinyOS) 환경구축
- FTDI 드라이버 설치 - 1



Windows 기반 TinyOS 환경 구축

- Cygwin K-mote(TinyOS) 환경구축
- FTDI 드라이버 설치 - 2



Windows 기반 TinyOS 환경 구축

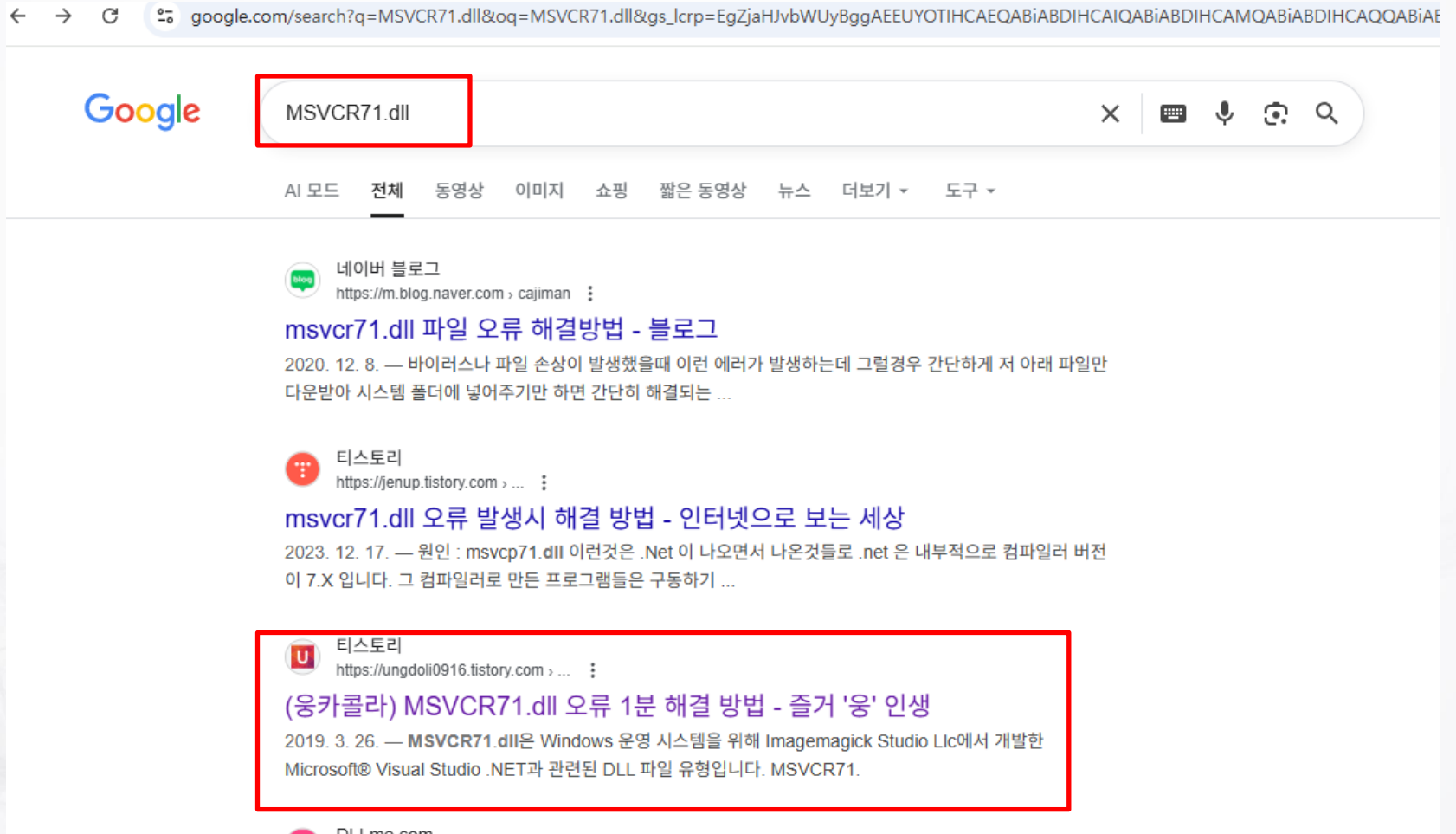
- Cygwin K-mote(TinyOS) 환경구축
- MSVCR71.dll 오류

```
/opt/tinyos-2.x/apps/Blink
-target=telosb -fnesc-cfile=build/telosb/app.c -board= -DDEFINED_TOS_AM_GROUP=0x
22 -DIDENT_APPNAME="BlinkAppC" -DIDENT_USERNAME="PC" -DIDENT_HOSTNAME="4-40
5-███" -DIDENT_USERHASH=0x6f10f836L -DIDENT_TIMESTAMP=0x68cb6afeL -DIDENT_UI
DHASH=0x7035cf12L BlinkAppC.nc -lm
    compiled BlinkAppC to build/telosb/main.exe
        2538 bytes in ROM
        56 bytes in RAM
msp430-objcopy --output-target=ihex build/telosb/main.exe build/telosb/main.ihex
    writing TOS image
tos-set-symbols --objcopy msp430-objcopy --objdump msp430-objdump --target ihex
build/telosb/main.ihex build/telosb/main.ihex.out-1 TOS_NODE_ID=1 ActiveMessageA
ddressC__addr=1
Could not find symbol ActiveMessageAddressC__addr in build/telosb/main.exe, igno
ring symbol.
Could not find symbol TOS_NODE_ID in build/telosb/main.exe, ignoring symbol.
    found mote on COM3 (using bsl,auto)
    installing telosb binary using bsl
tos-bsl --telosb -c 2 -r -e -l -p build/telosb/main.ihex.out-1
C:/cygwin64/bin/tos-bsl.exe: error while loading shared libraries: MSVCR71.dll:
cannot open shared object file: No such file or directory
make: *** [/opt/tinyos-2.x/support/make/msp/bsl.extra:46: program] Error 127

PC@4-405-███ /opt/tinyos-2.x/apps/Blink
$
```

Windows 기반 TinyOS 환경 구축

- Cygwin K-mote(TinyOS) 환경구축
- MSVCR71.dll 오류 - 1



Windows 기반 TinyOS 환경 구축

- Cygwin K-mote(TinyOS) 환경구축
- MSVCR71.dll 오류 - 2

하지만 위와 같은 방법으로도 문제가 해결되지 않는 경우가 종종 있기에, 아직 문제 해결이 되지 않으셨다면 아래에서 안내드리는 두번째 방법을 이용해보시길 바랍니다.

■ 추가 프로그램 설치 안내

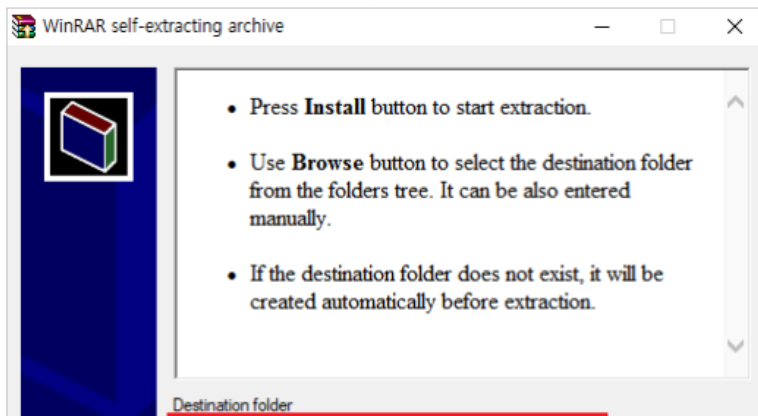
1. 아래의 첨부 파일을 클릭하여 다운로드한 후 실행시킵니다.

 m_71_auto.exe

2. PC의 운영체제에 맞게끔 아래의 경로를 복사하여 빨간색 네모 박스의 설치 경로를 수정 후 Install을 클릭하여 설치를 계속합니다.

- 32bit일 경우 : C:\Windows\System32

- 64bit일 경우 : C:\Windows\SysWOW64



- NesC 프로그래밍
 - ✓ <https://en.wikipedia.org/wiki/NesC>

nesC

🌐 6 languages ▼

Article Talk

Read Edit View history Tools ▼

From Wikipedia, the free encyclopedia



This article **does not cite any sources**. Please help improve this article by adding citations to reliable sources. Unsourced material may be challenged and removed.

Find sources: "NesC" – news · newspapers · books · scholar · JSTOR (March 2017) (Learn how and when to remove this template message)

This article is about the programming language. For the safety code, see National Electrical Safety Code.

nesC (pronounced "NES-see") is a [component-based, event-driven programming language](#) used to build applications for the [TinyOS](#) platform. TinyOS is an [operating environment](#) designed to run on embedded devices used in distributed [wireless sensor networks](#). nesC is built as an extension to the [C programming language](#) with components "wired" together to run applications on TinyOS. The name *nesC* is an abbreviation of "network embedded systems C".

Components and interfaces [edit]

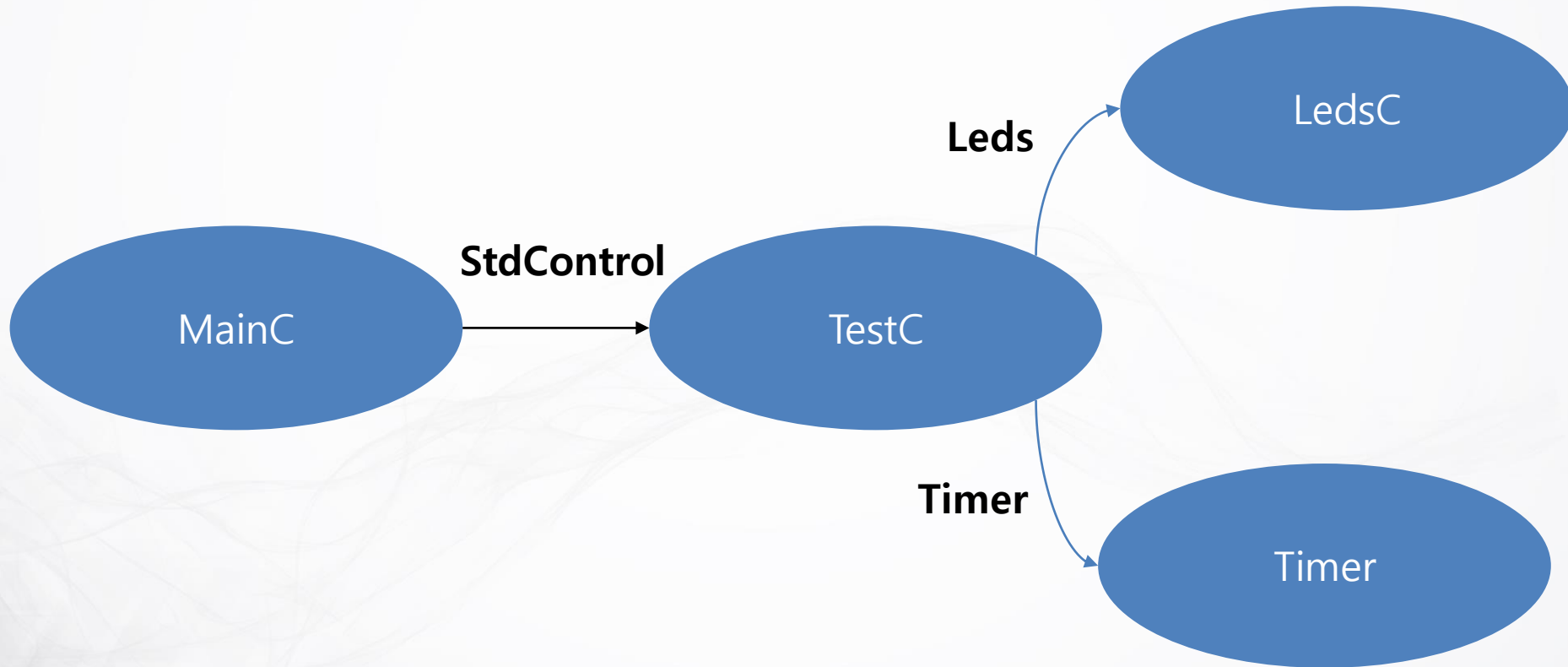
nesC programs are built out of components, which are assembled ("wired") to form whole programs. Components have internal concurrency in the form of tasks. Threads of control may pass into a component through its interfaces. These threads are rooted either in a task or a hardware interrupt.

Interfaces may be provided or used by components. The provided interfaces are intended to represent the functionality that the component provides to its user, the used interfaces represent the functionality the component needs to perform its job.

In nesC, interfaces are bidirectional: They specify a set of functions to be implemented by the interface's provider (commands) and a set to be implemented by the interface's user (events). This allows a single interface to represent a complex interaction between components (e.g., registration of interest in some event, followed by a callback when that event happens). This is critical because all lengthy commands in TinyOS (e.g. send packet) are non-blocking; their completion is signaled through an event (send done). By specifying interfaces, a component cannot call the send command unless it provides an implementation of the sendDone event. Typically commands call downwards, i.e., from application components to those closer to the hardware, while events call upwards. Certain primitive events are bound to hardware interrupts.

Components are statically linked to each other via their interfaces. This increases runtime efficiency, encourages robust design, and allows for better static analysis of programs.

- NesC 프로그래밍
 - ✓ Lesson1 : LED 1개 제어



- NesC 프로그래밍
 - ✓ BlinkAppC.nc

```
~/2020020/Blink
1 configuration BlinkAppC
2 {
3 }
4 implementation
5 {
6     components MainC, BlinkC, LedsC;
7     components new TimerMilliC() as Timer;
8
9
10    BlinkC -> MainC.Boot;
11
12    BlinkC.Timer -> Timer;
13    BlinkC.Leds -> LedsC;
14 }
15
```

- NesC 프로그래밍

- ✓ BlinkC.nc

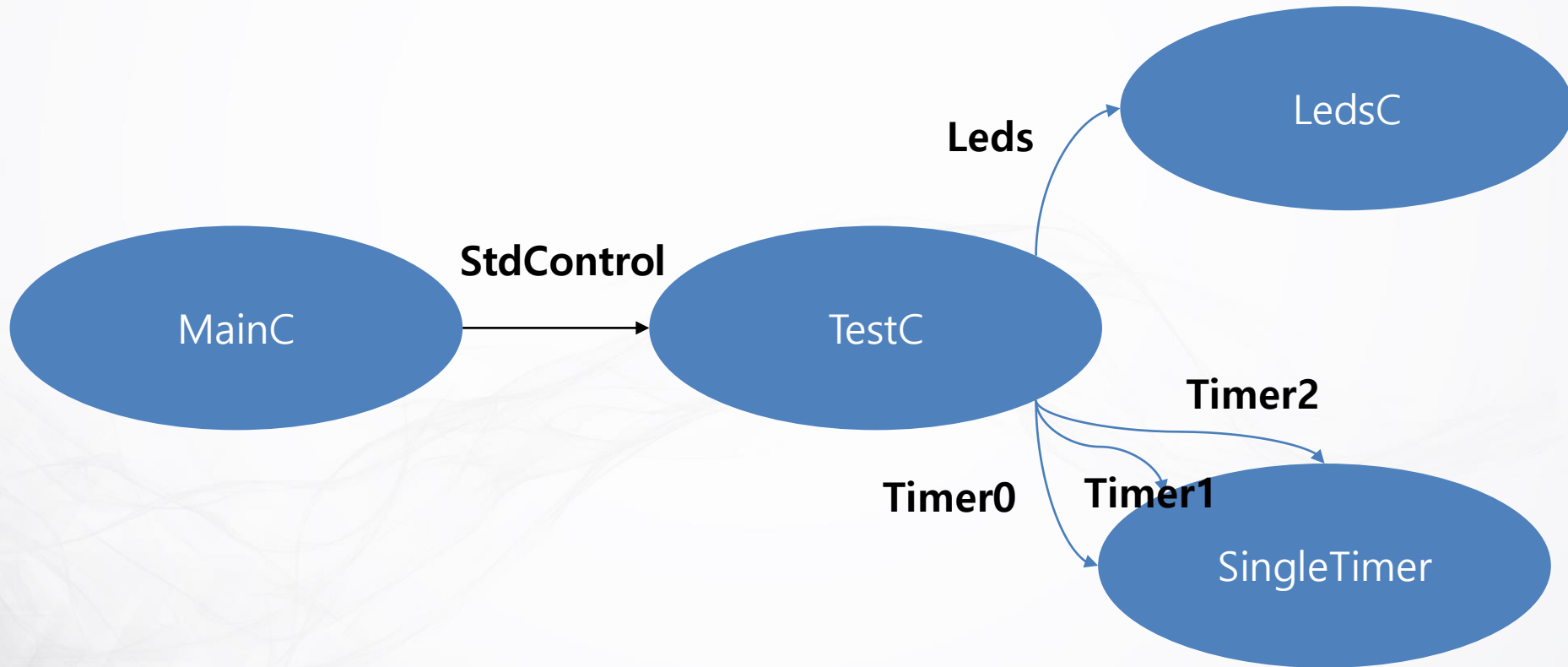
```
1 module BlinkC @safe()
2 {
3   uses interface Timer<TMilli> as Timer;
4   uses interface Leds;
5   uses interface Boot;
6 }
7 implementation
8 {
9   event void Boot.booted()
10  {
11    call Timer.startPeriodic( 250 );
12  }
13
14  event void Timer.fired()
15  {
16    call Leds.led0Toggle();
17  }
18
19 }
```

- NesC 프로그래밍
 - ✓ Makefile

```
1 COMPONENT=BlinkAppC
2 include $(MAKERULES)
3
~
```

- 컴파일 : \$ make telosb
- 퓨징 : \$ make telosb install.<노드아이디>

- NesC 프로그래밍
 - ✓ Lesson2 : LED 3개 제어



- NesC 프로그래밍
 - ✓ BlinkAppC.nc






```
50 configuration BlinkAppC
51 {
52 }
53 implementation
54 {
55     components MainC, BlinkC, LedsC;
56     components new TimerMilliC() as Timer0;
57     ?
58
59
60     BlinkC -> MainC.Boot;
61
62     BlinkC.Timer0 -> Timer0;
63     ?
64
65     BlinkC.Leds -> LedsC;
66 }
```


- NesC 프로그래밍

- ✓ BlinkC.nc

```
50 module BlinkC @safe()
51 {
52   uses interface Timer<TMilli> as Timer0;
53   ?
54   ?
55   uses interface Leds;
56   uses interface Boot;
57 }
58 implementation
59 {
60   event void Boot.booted()
61   {
62     call Timer0.startPeriodic( 250 );
63     ?
64   }
65 }
66
67 event void Timer0.fired()
68 {
69   dbg("BlinkC", "Timer 0 fired @ %s.Wn", sim_time_string())
```

- Github 소스코드 다운
- <https://github.com/sonnonet/inhatc>

	sonnonet Delete TestC.nc	f1be3ec 1 minute ago	🕒 43 commits
	2021_tcp_ip	Create docker-compose.yml	2 years ago
	A반/96s	Create README.MD	2 years ago
	BaseStation_v1.1	Delete oscilloscope.py	7 minutes ago
	TestLowOneHopSht_sc	Delete TestC.nc	1 minute ago

TestAppC.nc

```
~/sonnonet_tinyos/Tinyos/TestLowOneHopSht_sc
1 includes Test;
2 configuration TestAppC
3 {
4 }
5 implementation
6 {
7   components TestC, MainC;
8   components LedsC, new TimerMilliC();
9
10  components ActiveMessageC as AMC;
11  components new AMSenderC(AM_TEST_DATA_MSG) as AMSC;
12
13  TestC.Boot -> MainC;
14  TestC.Leds -> LedsC;
15  TestC.MilliTimer -> TimerMilliC;
16
17  TestC.RadioControl -> AMC;
18  TestC.RadioSend -> AMSC;
19
20  components new SensirionSht11C() as Sht11Ch0C;
21  TestC.Temp -> Sht11Ch0C.Temperature;
22  TestC.Humi -> Sht11Ch0C.Humidity;
23
24  components new I11uAdcC() as I11u;
25  TestC.I11u -> I11u;
26
27  components BatteryC;
28  TestC.Battery -> BatteryC;
29 }
```

Test.h

~/sonnonet_tinyos/Tinyos/TestLowOneHopSht_sc

```
1 #ifndef TEST_H
2 #define TEST_H
3 #include "message.h"
4 enum {
5     TEST_PERIOD = 10240LU,
6 };
7 enum {
8     DFLT_VAL = 0x11,
9 };
10 enum {
11     TEST_DATA_LENGTH = TOSH_DATA_LENGTH - 6,
12 };
13 enum {
14     AM_TEST_DATA_MSG = 0xA4,
15 };
16
17 typedef nx_struct test_data_msg {
18     nx_am_addr_t srold;
19     nx_uint32_t seqNo;
20     nx_uint16_t type;
21     nx_uint16_t Temp;
22     nx_uint16_t Humi;
23     nx_uint16_t Illu;
24     nx_uint16_t battery;
25     //nx_uint8_t testData[TEST_DATA_LENGTH];
26 } test_data_msg_t;
27
28 #endif // TEST_H
```

TestC.nc

~/sonnonet_tinyos/Tinyos/TestLowOneHopSht_sc

```
1 module TestC
2 {
3   uses {
4     ?
5   }
6
7   interface SplitControl as RadioControl;
8   interface AMSend as RadioSend;
9
10  interface Read<uint16_t> as Temp;
11  interface Read<uint16_t> as Humi;
12  interface Read<uint16_t> as Illu;
13
14  interface Battery;
15 }
16 }
17 }
18 }
```

TestC.nc

~/sonnonet_tinyos/Tinyos/TestLowOneHopSht_sc

```
19 implementation
20 {
21     message_t testMsgBfrr;
22     test_data_msg_t *testMsg;
23
24     uint32_t seqNo;
25     uint8_t step;
26
27
28     task void startTimer();
29     event void Boot.booted() {
30         testMsg = (test_data_msg_t *)call RadioSend.getPayload(
31             &testMsgBfrr, sizeof(test_data_msg_t));
32         testMsg->srcID = TOS_NODE_ID;
33
34         seqNo = 0;
35
36         post startTimer();
37     }
38
39     task void startTimer() {
40         call MilliTimer.startPeriodic(TEST_PERIOD);
41     }
42
43     task void radioOn();
44     event void MilliTimer.fired() {
45         post radioOn();
46     }
47 }
```

TestC.nc

```
47
48 void startDone();
49 task void radioOn() {
50     if (call RadioControl.start() != SUCCESS) startDone();
51 }
52
53 event void RadioControl.startDone(error_t error) {
54     startDone();
55 }
56
57 task void readTask();
58 void startDone() {
59     step = 0;
60     post readTask();
61     call Leds.led0Toggle();
62 }
63
64 void sendDone();
65 task void sendTask() {
66     testMsg->seqNo = seqNo++;
67     testMsg->type = 2; //THL type 2
68
69     if (call RadioSend.send(AM_BROADCAST_ADDR, &testMsgBfrr,
70         sizeof(test_data_msg_t)) != SUCCESS) sendDone();
71     call Leds.led2Toggle();
72 }
73
```

TestC.nc

```
70 |
74 | event void RadioSend.sendDone(message_t* msg, error_t error) {
75 |     sendDone();
76 | }
77 |
78 | task void radioOff();
79 | void sendDone() {
80 |     call Leds.led00ff();
81 |     call Leds.led10ff();
82 |     call Leds.led20ff();
83 |     post radioOff();
84 | }
85 |
86 | void stopDone();
87 | task void radioOff() {
88 |     if (call RadioControl.stop() != SUCCESS) stopDone();
89 | }
90 |
91 | event void RadioControl.stopDone(error_t error) {
92 |     stopDone();
93 | }
94 |
95 | void stopDone() {
96 | }
97 |
```


TestC.nc

```
97  task void readTask() {
98      switch(step) {
99          case 0:
100              call Temp.read(); break;
101          case 1:
102              call Humi.read(); break;
103          case 2:
104              call Illu.read(); break;
105          default:
106              testMsg->battery = call Battery.getVoltage();
107              post sendTask();
108              break;
109      }
110      step += 1;
111  }
112
113  event void Temp.readDone(error_t error, uint16_t val) {
114      //if (error != SUCCESS) call Leds.led00n();
115      testMsg->Temp = error == SUCCESS ? val : 0xFFFA;
116      post readTask();
117  }
118  event void Humi.readDone(error_t error, uint16_t val) {
119      //if (error != SUCCESS) call Leds.led10n();
120      testMsg->Humi = error == SUCCESS ? val : 0xFFFB;
121      post readTask();
122  }
123  event void Illu.readDone(error_t error, uint16_t val){
```

TestC.nc

```
123  event void Illu.readDone(error_t error, uint16_t val){
124      testMsg->Illu = error == SUCCESS ? val : 0xFFFC;
125      post readTask();
126  }
127 }
```

- 퓨징

```
SUPER@DESKTOP-4EUDDMP ~/2023_inhatc/Zigbee/TestLowOneHopSht_sc
$ ls
Battery      BatteryC.nc  Makefile      chips_sht11    tos.pyc
Battery.h    IlluAdcC.nc  Msp430RefVoltGenerator.h  platforms_sht11
Battery.nc   IlluAdcP.nc  Msp430RefVoltGeneratorP.nc  tos.py
SUPER@DESKTOP-4EUDDMP ~/2023_inhatc/Zigbee/TestLowOneHopSht_sc
$ make telosb install.노드아이디|
```

NesC 미들웨어 프로그래밍 (온습도센서 파싱)

oscilloscope.py

```
# Data Format
# CO2 Data0
# THL Temperature Data0, Humidity Data1, Illumination Data2, Battery Data3

import sys
import tos
import datetime
import threading

AM_OSCILLOSCOPE = 0x93

class OscilloscopeMsg(tos.Packet):
    def __init__(self, packet = None):
        tos.Packet.__init__(self,
                               [('srcID', 'int', 2),
                                ('seqNo', 'int', 4),
                                ('type', 'int', 2),
                                ('Data0', 'int', 2),
                                ('Data1', 'int', 2),
                                ('Data2', 'int', 2),
                                ('Data3', 'int', 2),
                                ('Data4', 'int', 2)],
                               packet)

if '-h' in sys.argv:
    print "Usage:", sys.argv[0], "serial@/dev/ttyUSB0:57600"
    sys.exit()

am = tos.AM()

while True:
    p = am.read()
    msg = OscilloscopeMsg(p.data)
```

NesC 미들웨어 프로그래밍 (온습도센서 파싱)

```
##### CO2 Logic #####
if msg.type == 1:
#   pbr_Num = pbrNum_return(msg.srcID)
   pbr_Num = 3
   CO2 = msg.Data0
   CO2 = 1.5 * CO2 / 4096 * 2 * 1000
   ### MySQL Insert ###
   try:
       with conn.cursor() as curs:
           Now = datetime.datetime.now()
           sql = """insert into JB_Sensor_CO2(NODE_ID,SEQ,CO2,PBR_NUM,REGDATE)
                       values(%s, %s, %s, %s, %s)"""
           curs.execute(sql, (msg.srcID,msg.seqNo,CO2,pbr_Num,Now))
           conn.commit()
   except:
       conn.close()
   print "ID:",msg.srcID, "seqNo:",msg.seqNo, "CO2:",CO2

##### THL Logic #####
if msg.type == 2:
    battery = msg.Data3
    #   battery = 0

    Illumi = int(msg.Data2)
    humi = -2.0468 + (0.0367*msg.Data1) + (-1.5955*0.000001)*msg.Data1*msg.Data1
    temp = -(39.6) + (msg.Data0 * 0.01)

    print "id:" , msg.srcID, " Count : ", msg.seqNo, \
          "Temperature: ",temp, "Humidity: ",humi, "Illumination: ",Illumi, "Battery : ", battery

##### PH Sensor Logic #####
if msg.type == 3:
    #hex1 = hex(msg.Data2)
    #hex1 = hex1[:-1]
    #hex2 = hex(msg.Data2_1)
    #hex2 = hex2[:-1]
    #   print "before : ", PH_str
    #   print (int(hex1,0)), (int(hex2,0))
    #   print hex(msg.Data2), hex(msg.Data2_1)

    PH = msg.Data0
```