# 아두이노와 라즈베리파이 시리얼 통신하기

http://www.notforme.kr/archives/756

[2013/10/11](http://www.notforme.kr/archives/756)  [jwj0831](http://www.notforme.kr/archives/author/jwj0831)  [IT](http://www.notforme.kr/archives/category/it)

대학원 수업 프로젝트 아이디어로 위 슬라이드의 실내 먼지측정 시스템을 제안했습니다. 아이디어를 제안하면서 전부터 써보고 싶었던 아두이노와 라즈베리파이를 활용하겠다는 계획을 담았지요. 그 덕에 오늘 수업 재료비로 이 두녀석을 드디어 받았습니다. 하하핫!  
[](http://www.notforme.kr/jthkhhjwj/wp-content/uploads/2013/10/2013-10-11-16.09.14.jpg)

아두이노를 활용하는 것은 그리 어렵지 않고, 라즈베리파이도 리눅스 사용이 익숙하신 분들은 어렵지 않게 기본 셋팅을 각각의 공식 사이트에 들어가서 확인할 수 있답니다. 저도 구매한 microSD 카드에 라즈비안 OS 이미지를 심고 기본 셋팅(비번설정, 패키지 업데이트)을 먼저 했습니다.

드디어 본래 계획한 프로젝트대로 아두이노와 라즈베리파이의 간단한 연결테스트를 시작합니다. 참고는 거의 전적으로 이 블로그입니다.

http://blog.oscarliang.net/connect-raspberry-pi-and-arduino-usb-cable/

제가 한글로 적었다는 것과 몇가지 차이점 빼고는 동일합니다.

일단 아두이노에서는 다음과 같이 시리얼 연결을 하고 정해진 시간마다 “Hello Pi”라는 문자열을 날리는 코드를 다음과 같이 작성합니다.



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | void setup(){    Serial.begin(9600);  }    void loop(){    Serial.println("Hello Pi");    delay(1000);  } |

C코드 같으면서도 딱 봐도 무슨 의미인지 들어오지요? 이래서 비전공자도 쉽게 아두이노를 가지고 놀 수 있다는 거지요 ^^ 자 이제 라즈베리파이 녀석에서는 파이썬으로 간단하게 시리얼 연결 코드를 작성합니다.



|  |  |
| --- | --- |
| 1  2  3  4  5  6 | import serial    ser = serial.Serial('/dev/ttyACM0', 9600)    while 1:      print ser.readline() |

아주 간단한 코드이지요. 시리얼 설정과 함께 ser객체 생성 후 while 루프 안에서 계속 데이터를 받아서 출력하겠다는 것이지요. 이대로 따라하면 잘 될 것 같은데 위 코드를 실행하면 에러가 뜹니다. 왜냐하면 serial 모듈이 이 라즈베리 파이 안에 없기 때문이지요. 따라서 python의 pip를 활용해서 serial 모듈을 다운받아야 정상적으로 작동합니다.



|  |  |
| --- | --- |
| 1  2 | $ sudo apt-get install python-pip  $ sudo pip install pyserial |

요렇게 serial 모듈까지 설치하고나면 짜잔

[](http://www.notforme.kr/jthkhhjwj/wp-content/uploads/2013/10/2013-10-11-18.57.40.jpg)

사진과 같이 모니터에 “Hello Pi”가 거침없이 찍혀 내려오는 모습을 보실 수 있습니다. 와우!~ 눈물이 날 뻔…. 아니 그 정도의 감동은 아니었지만 “Hello Pi”가 모니터에 찍힐 때 감동은 C를 처음 배울 때 “Hello World”와 비견될 정도의 감동이네요.

간간히 프로젝트를 진행하면서 아두이노와 라즈베리파이 관련 포스팅을 하겠습니다. 아! 그리고 라즈베리파이보다 스펙이 좀 더 좋은 [비글본 블랙](http://www.leocom.kr/Beaglebone/Default.aspx)을 체험단으로 무상으로 곧 받게되어 비글본 블랙을 가지고도 포스팅을 올릴 예정입니다. ^^

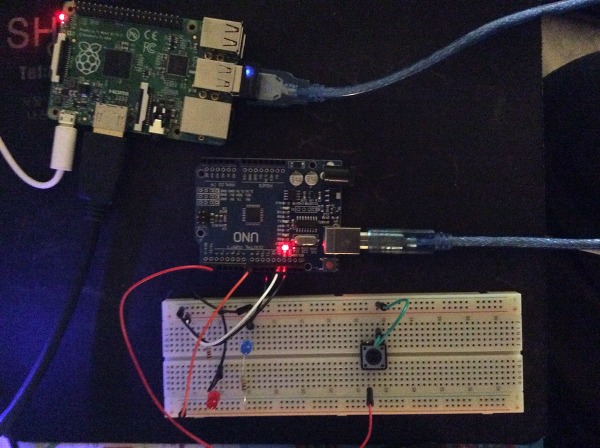
<http://mykumi.tistory.com/m/post/17>

## [라즈베리파이 + 아두이노 Serial 연결](http://mykumi.tistory.com/m/post/17)

마이지연 | 2015/03/29 22:23 | [raspberry pi](http://mykumi.tistory.com/m/post?categoryId=832649)앱으로 보기

라즈베리파이와 아두이노를 시리얼 케이블로 연결하여 라즈베리파이에 설치된 Node.js 기반의 웹화면에서 LED를 켜거나 끄도록 하는 간단한 테스트를 진행합니다.

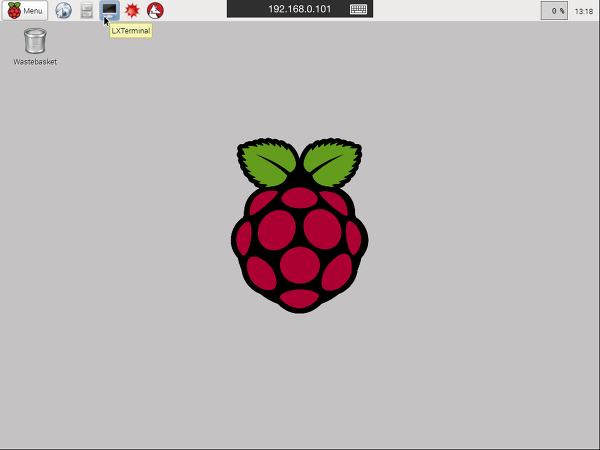
**1. 아두이노와 라즈베리파이의 시리얼 포트 연결 모습**



빵판에 몇가지 연결되어 있지만 13번 Pin에 연결된 붉은색 LED를 사용합니다.

**2. Arduino 개발환경 설치**

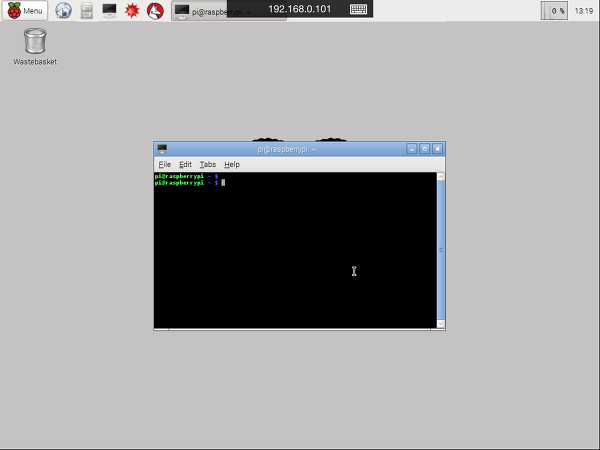
1) GUI 환경에서 LX Terminal을 실행합니다.



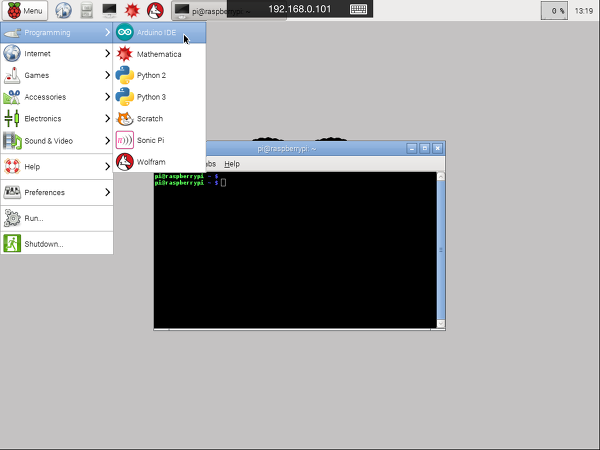
2) 아두이노 개발환경을 apt-get으로 설치하고 시리얼 연결을 위한 권한 부여

// 아두이노 개발환경 설치  
$ sudo apt-get install arduino

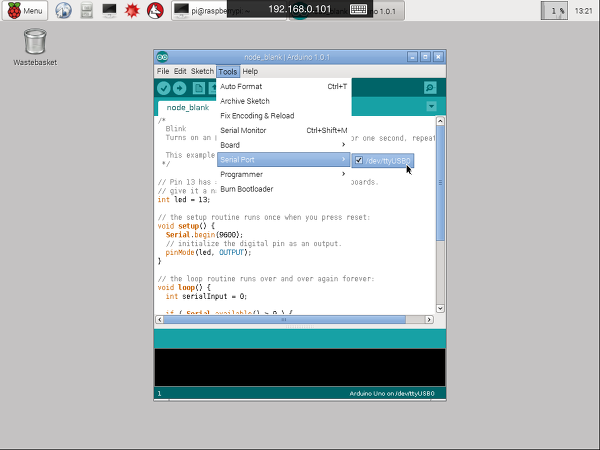
// pi 계정을 시리얼 통신을 위한 권한그룹에 추가  
$ sudo usermod -a -G tty pi  
$ sudo usermod -a -G dialout pi



3) 아두이노 개발환경 : Arduino IDE 실행



4) Serial Port 설정



※ 해당 Serial Port는 나중 프로그램에 사용되니 외워둡시다 ^^

**3. LED 제어를 위한 프로그램 작성 및 Upload**

int led = 13;  
int serialInput = 0;

void setup() {          
  pinMode(led, OUTPUT);       
  Serial.begin(9600);    
}

void loop() {  
  if ( Serial.available() > 0 ) {  
    serialInput = Serial.read();  
  }

  if (serialInput == 49) {  
    digitalWrite(led, HIGH);     
  }

  if (serialInput == 48) {  
    digitalWrite(led, LOW);      
  }  
}

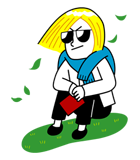
일단 라즈베리 파이에 개발환경을 설치하고 아두이노와 연결하여 프로그램을 작성 및 업로드하는 부분까지 수행하였습니다.

다음에는 라즈베리파이에 Node.js와 Express를 설치하여 웹화면에서 아두이노를 제어하는 부분을 진행하도록 하겠습니다.

<http://m.blog.naver.com/agapeuni/220198811728>

**라즈베리 파이(Raspberry Pi) 와**

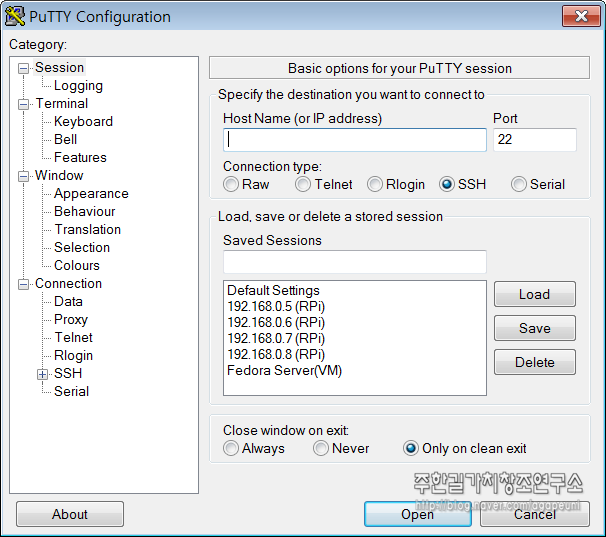
**아두이노 우노(Arduino UNO)**



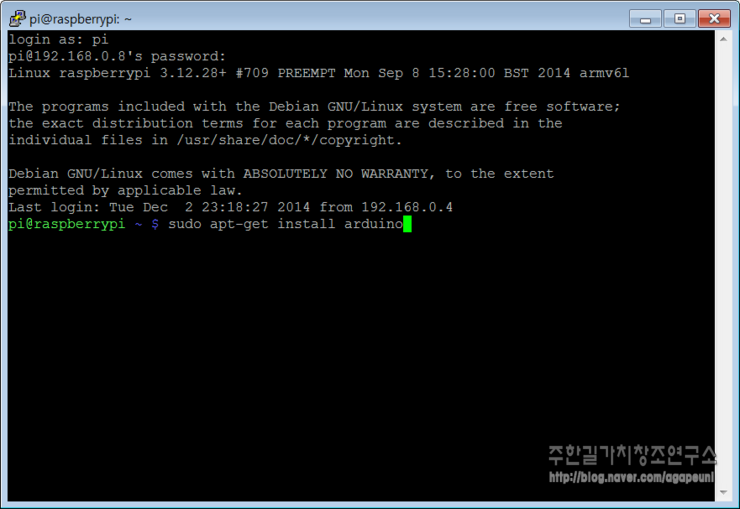
안녕하세요. 가치창조 주한길입니다.

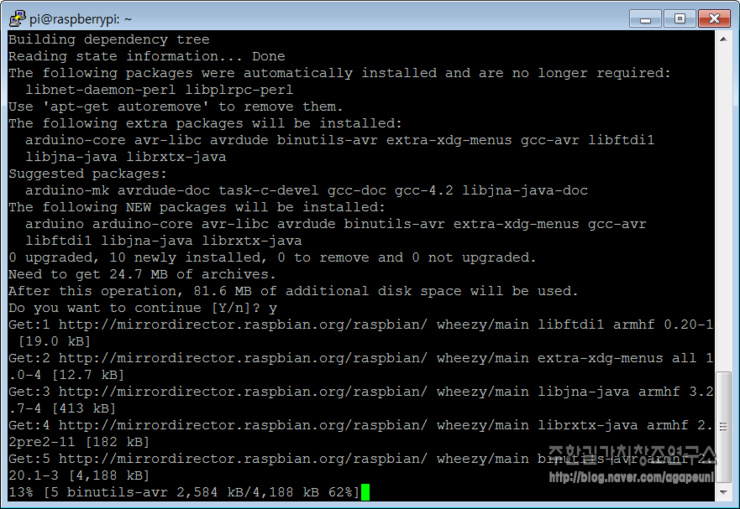
라즈베리 파이에 아두이노를 설치하고 아두이노에 코드를 업로드하여 라즈베리 파이와 간단한 통신을 하는 실험입니다.

먼저 PuTTY를 사용하여 SSH로 **라즈베리 파이(Raspberry Pi)**에 접속을 합니다.

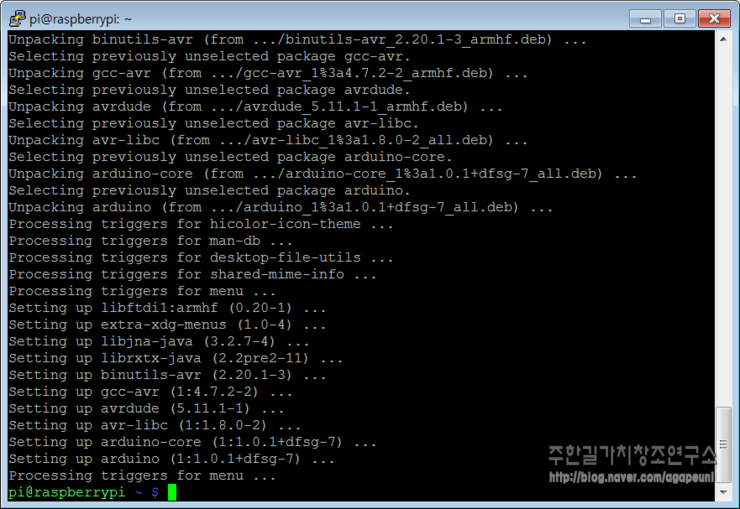


**라즈베리 파이(Raspberry Pi)**에서 **아두이노 우노(Arduino UNO)**를 제어하기 위해 arduino 패키지를 설치합니다.

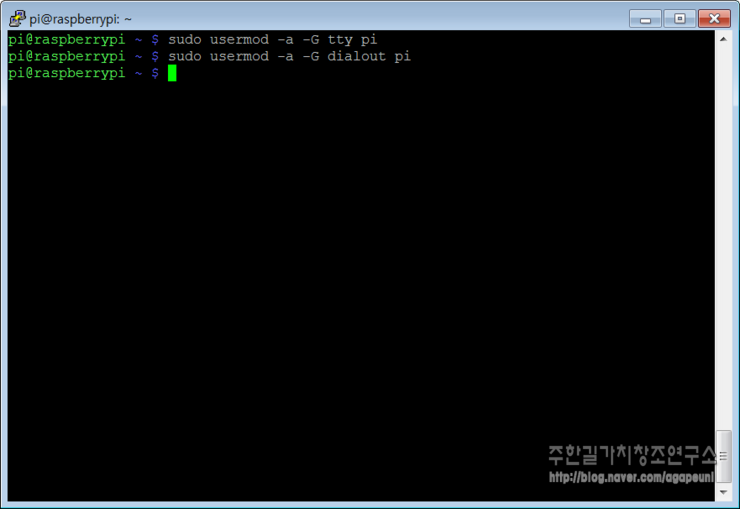
  
  
  
다른 패키지 설치에 비교하면 상당한 로그가 출력되어 지나갑니다.



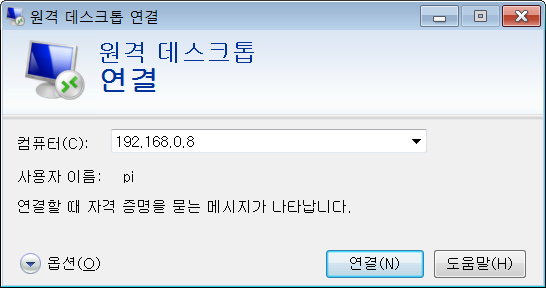
arduino 패키지가 성공적으로 설치되었습니다.



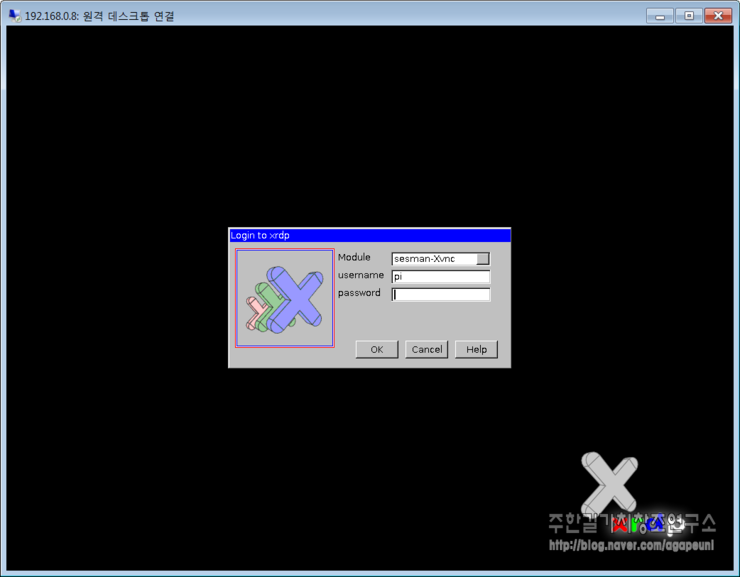
LXDE에서 arduino에 접근하기 위해 아래와 같이 pi 계정에 tty dialout 그룹을 지정합니다.



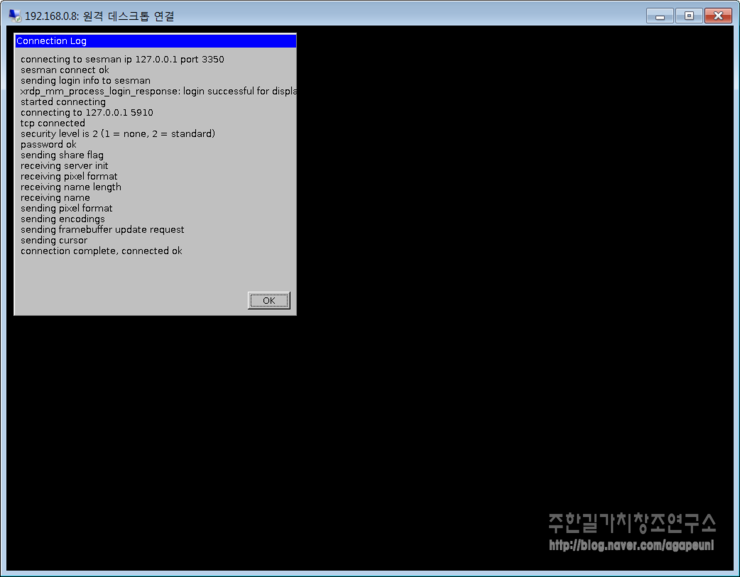
LXDE에서 작업하기 위해 윈도우에서 **라즈베리 파이(Raspberry Pi)**에 원격 데스크톱으로 접속합니다.



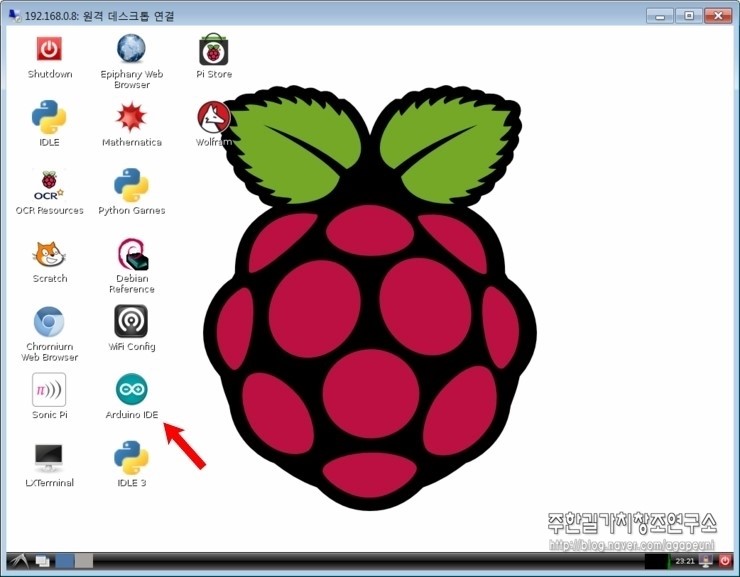
Login to xrdp 창에 로그인 계정과 비밀번호를 입력합니다.



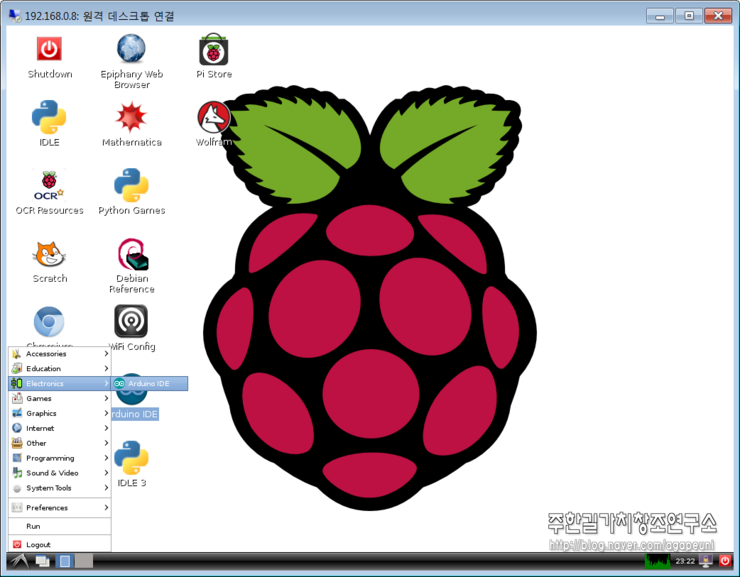
접속 로그창에 성공적인 접속정보가 표시됩니다.



원격으로 LXDE에 연결되었습니다. 바탕화면에 보면 Arduino IDE 아이콘이 새롭게 추가되어 있습니다.



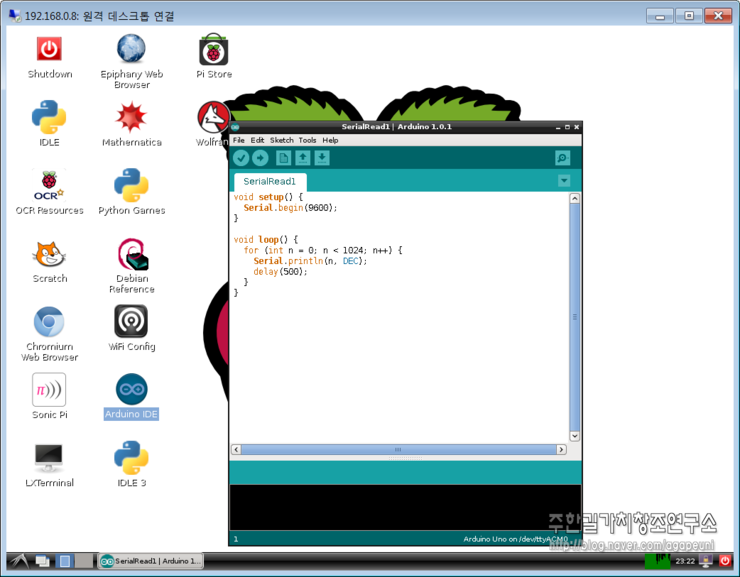
시작 - Electronics - Arduino IDE 를 클릭하여  Arduino IDE를 실행합니다.



**실험1. 아두이노 우노 소스코드 (SerialRead1.ino)**

**아두이노 우노(Arduino UNO)**에서 시리얼 통신으로 0부터 1023까지 값을 0.5초 간격으로 출력하는 코드입니다.

|  |
| --- |
| void setup() {      Serial.begin(9600);  }    void loop() {      for (int n = 0; n < 1024; n++) {          Serial.println(n, DEC);          delay(500);      }  } |



**실험1.  라즈베리 파이 소스코드 (SerialRead1.py)**

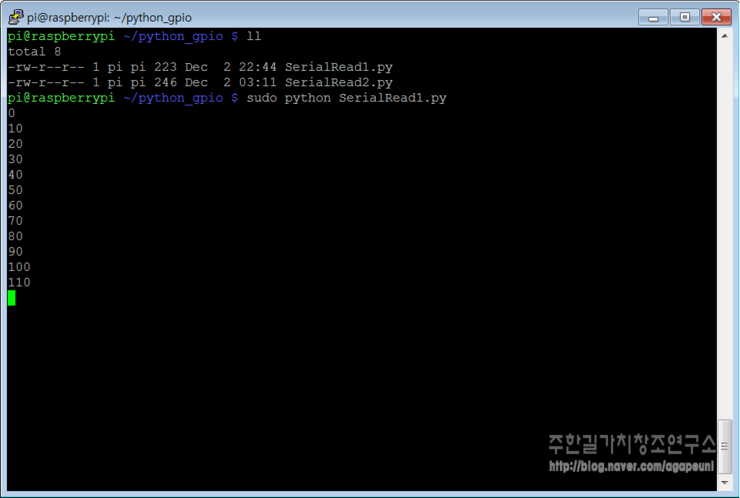
**아두이노 우노(Arduino UNO)**에서 시리얼 통신으로 보낸 값을

**라즈베리 파이(Raspberry Pi)**에서 반복하여 읽어서 10을 곱해서 출력하는 코드입니다.

|  |
| --- |
| import serial    port = "/dev/ttyACM0"  serialFromArduino = serial.Serial(port, 9600)  serialFromArduino.flushInput()  while True:          input\_s = serialFromArduino.readline()          input = int(input\_s)          print(input \* 10) |

PuTTY(SSH)로 **라즈베리 파이(Raspberry Pi)**에 접속한 화면에서 아래와 같이 SerialRead1.py를 실행하면

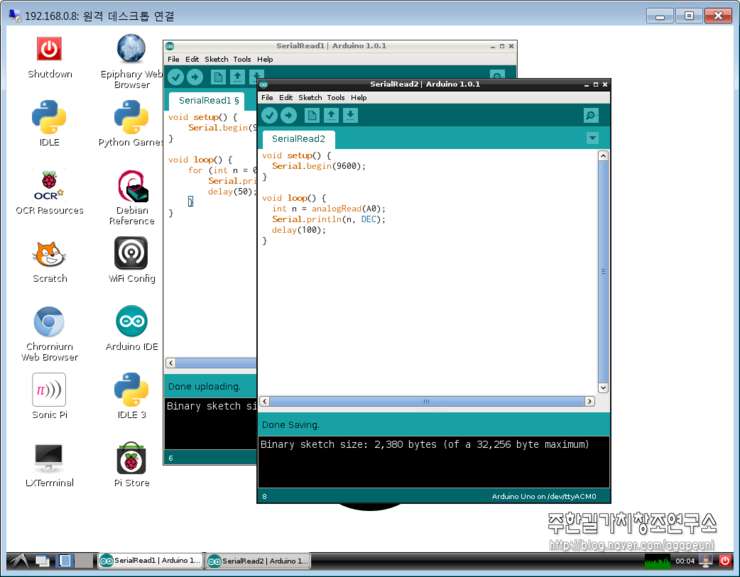
**아두이노 우노(Arduino UNO)**에서 전송된 값에 계산된 (\*10) 값이 화면에 순서대로 표시됩니다.



**실험2. 아두이노 우노 소스코드 (SerialRead2.ino)**

**아두이노 우노(Arduino UNO)**의 A0핀에서 아날로그 값을 읽어서 시리얼 통신으로 출력합니다.

|  |
| --- |
| void setup() {      Serial.begin(9600);  }    void loop() {      int n = analogRead(A0);      Serial.println(n, DEC);      delay(100);  } |



**실험2.  라즈베리 파이 소스코드 (SerialRead1.py)**

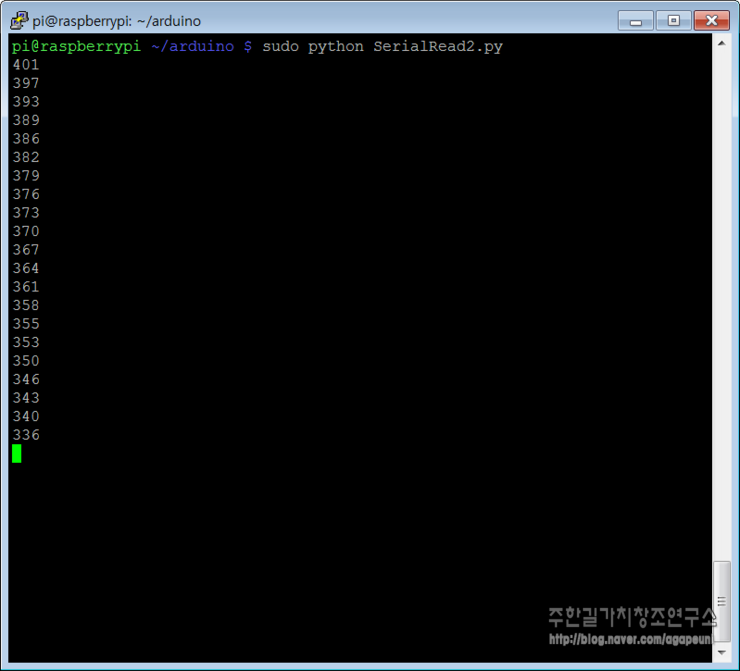
**아두이노 우노(Arduino UNO)**에서 시리얼 통신으로 출력하여 보낸 값을

**라즈베리 파이(Raspberry Pi)**에서 반복하여 읽어서 바로 출력하는 코드입니다.

|  |
| --- |
| import serial    port = "/dev/ttyACM0"  serialFromArduino = serial.Serial(port, 9600)  serialFromArduino.flushInput()  while True:          input\_s = serialFromArduino.readline()          input = int(input\_s)          print(input) |

임의의 값이 일정한 범위내에서 무작위로 표시되고 있습니다.

A0 값을 씨드로 해서 랜덤 함수를 만들 때 자주 사용하곤 합니다.

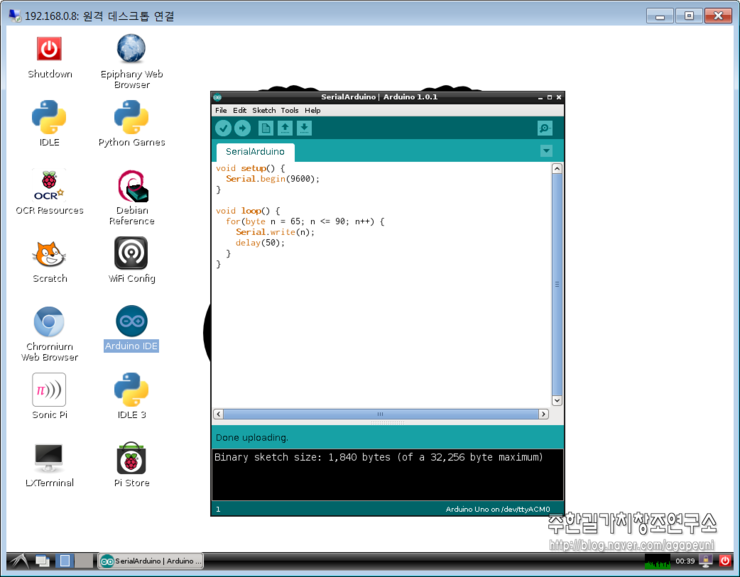


**실험3. 아두이노 우노 소스코드 (SerialArduino.ino)**

**아두이노 우노(Arduino UNO)**에서 시리얼 통신으로 65부터 90까지 값을 0.05초 간격으로 출력하는 코드입니다.

65는 ASCII 값으로 A를 나타내고 90은 Z를 나타냅니다.

|  |
| --- |
| void setup() {     Serial.begin(9600); }  void loop() {     for(byte n = 65; n <= 90; n++) {         Serial.write(n);         delay(50);     } } |



**실험3.  라즈베리 파이 소스코드 (SerialArduino.py)**

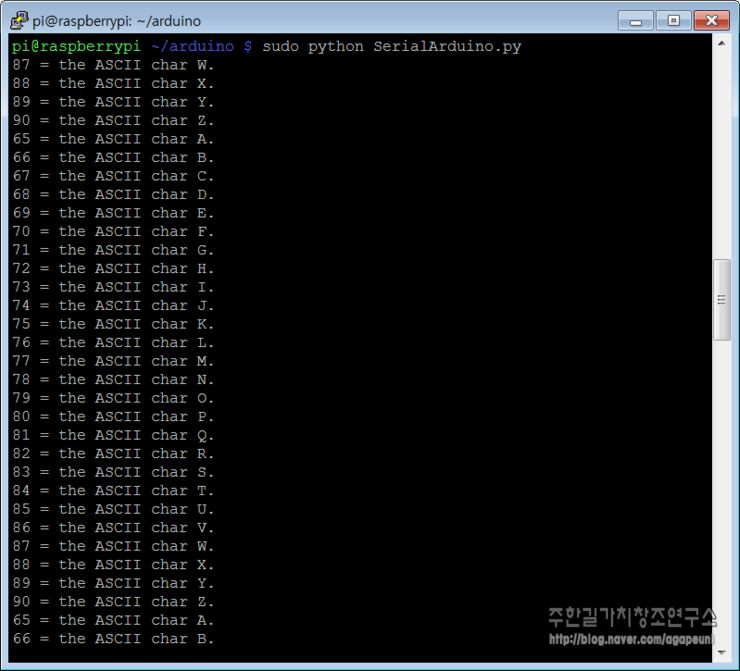
**아두이노 우노(Arduino UNO)**에서 시리얼 통신으로 보낸 값을

**라즈베리 파이(Raspberry Pi)**에서 하나씩 읽어서 10진수 값과 ASCII 문자로 화면에 표시합니다.

|  |
| --- |
| import serial  port = "/dev/ttyACM0" serialArduino = serial.Serial(port, 9600) serialArduino.flushInput() while True:     if (serialArduino.inWaiting() > 0):          input = serialArduino.read(1)          print(str(ord(input)) + " = the ASCII char " + input + ".") |

아래와 같이 SerialArduino.py를 실행하면  **아두이노 우노(Arduino UNO)**에서 전송된 값에 화면에 출력됩니다.

65부터 90까지 반복하여 출력합니다. ASCII로는 A부터 Z까지 순서대로 표시됩니다.



**[로마서 6장 5절 말씀]**

**만일 우리가 그의 죽으심과 같은 모양으로 연합한 자가 되었으면**

**또한 그의 부활과 같은 모양으로 연합한 자도 되리라**