

Getting started met Arduino Yun!!!

Samenvatting Arduino Yun

Inhoudsopgave

1. Waarom Arduino Yun?
2. Beschrijving van de Arduino Yun hardware
3. Resetten van de Arduino Yun en verschillende onderdelen
4. Atmel ATmega32U4
5. Drivers voor de Arduino Yun installeren
6. Onboard Wifi configureren
7. Reset het password van jouw Arduino Yun
8. Een virtuele serial port
9. Programmeren met Arduino IDE via wireless connectie met Arduino Yun
10. Programmeren met Arduino IDE via ethernet connectie met de Arduino Yun
11. Communicatie tussen Arduino en OpenWrt-Yun via Bridge library
 - 11.1 De Console
 - 11.2 Process
 - 11.3 Bridge gebruiken om informatie tussen de twee processoren uit te wisselen
12. AR9331 met OpenWrt-Yun
 - 12.1 Wireless en systeem configuratie
 - 12.2 Installeren van software (packages)
 - 12.3 Communiceren met/tussen OpenWrt-Yun/Yun deel en computer
 - 12.4 CLI
 - 12.5 Upgrade de OpenWRT versie op jouw Yun
 - 12.6 Externe opslag op de Yun (microSD card)
13. Python
14. Acknowledgement

1. Waarom Arduino Yun?

Redenen voor mijzelf om een Arduino Yun uit te proberen zijn:

- Onboard Wifi
- Onboard Micro-SD kaart opslag (extra ruimte) (ik liep uit geheugen met de Uno met functies en internet connecties)
- Meer mogelijkheden in scripting en programmeren met Linux, Python, etc. (ik liep uit geheugen met de Uno)

2. Beschrijving van de Arduino Yun hardware

De Arduino Yun heeft twee processoren:

- Atmel ATmega32U4 (de arduino processor/chip)
- Atheros AR9331 (Linux en OpenWrt wireless stack, plus full install van Python 2.7)

Andere zaken op het bordje:

- MicroSD card slot
- USB A of USB host connector
- Micro-USB (voor connectie met de computer en stroomvoorziening)

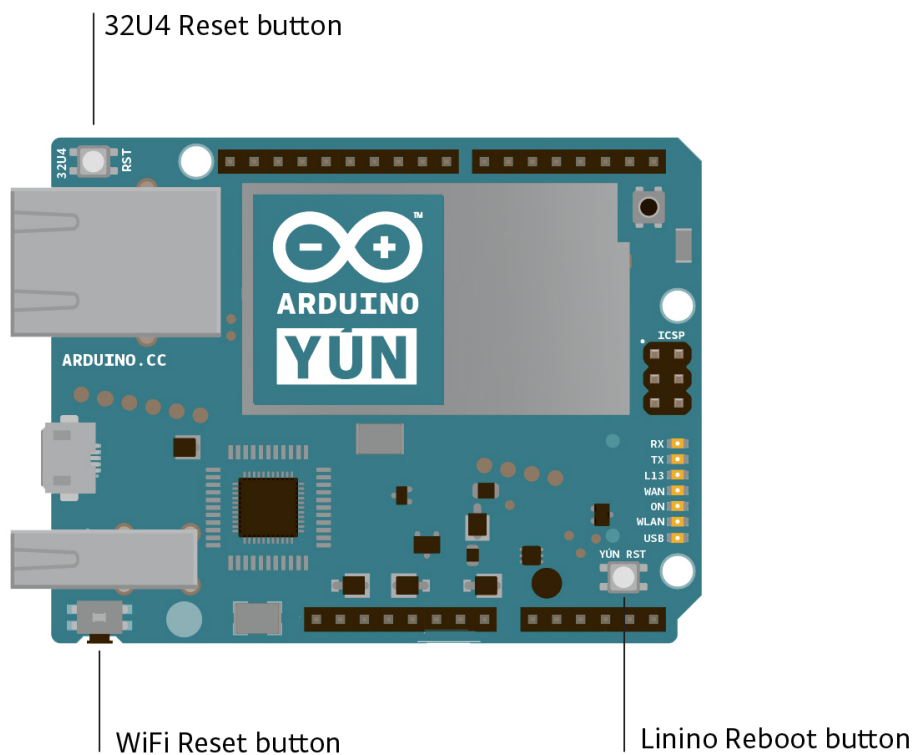
- Ethernet connectie
- Integrated Wifi module (connect to wireless router of gebruik als access point).

De Arduino Yun gebruikt een 5V stroomvoorziening. Er is geen 5V power regulator aan boord dus zorg dat er nooit een stroombron van meer dan 5V op aangesloten wordt, daarmee sloop je de Yun. Een alternatief voor de stroomvoorziening via Micro-USB en pc is de Vin, maar let daarbij op de 5V.

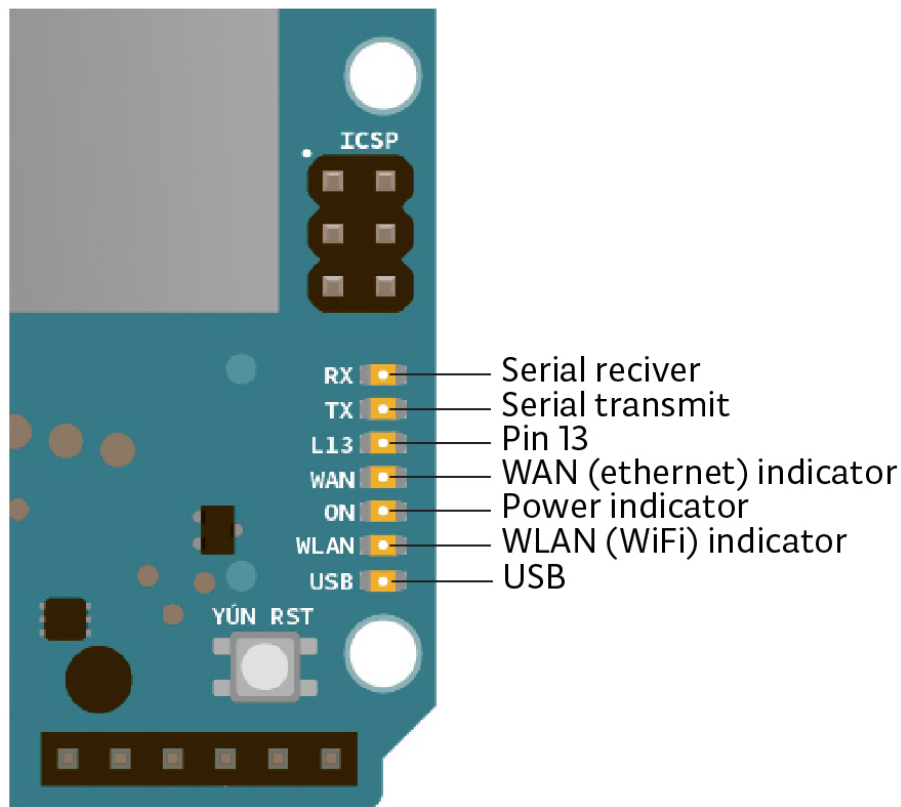
De SD, Ethernet en USB-A connectors zijn fysiek niet aangesloten op de Arduino processor (32U4) maar op de AR9331 (Linux deel).

De `Serial1` van de Arduino processor is gereserveerd voor communicatie met de AR9331 chip (via de Bridge library). Niet gebruiken.

De 32u4, WiFi, and AR9331 processors hebben allen hun eigen reset buttons.



Status indicatoren op de Yun:



Voor meer informatie over de Arduino Yun hardware zie de: [Yún hardware page](#) (nog verder uitwerken).

3. Resetten van de Arduino Yun en verschillende onderdelen

Om het Linux deel te resetten (de AR9331 chip die OpenWrt-Yun boot) druk op YUN RST (dicht bij de analoge pins en LEDs op het bord).

Om de 32U4 te restarten (met de huidige Arduino sketch) druk **twee** keer op de knop naast de Ethernet poort.

Om de Wifi te resetten druk op de knop naast de USB-A connector met het label WLAN RST. Het WLAN LEDje gaat flashen (druk hem niet te lang in, zie hierna).

Om de Wifi te resetten om bijv. op een ander netwerk weer een connectie met de pc tot stand te brengen druk de WLAN RST knop langer dan 5 seconden in, maar korter dan 30. De AR9331 processor gaat dan rebooten. De Wifi configuratie wordt gereset en de Yun zal zijn eigen Wifi netwerk starten (Arduino Yun-xxxxxxxetc). Alle andere modificaties/configuraties zullen behouden blijven.

Om het OpenWrt-Yun deel te resetten naar default, druk op de WLAN RST knop voor langer dan 30 seconden. Het bord zal terugkeren in zijn originele settings alsof het uit de doos komt. Of met de laatste OpenWRT image erop als je deze eerder geflashed hebt. Dit

verwijderd alle geïnstalleerde files en netwerk settings.

4. Atmel ATmega32U4

- Te programmeren via Arduino IDE, via USB (micro) connectie of als je de Yun wireless hebt aangesloten via Wifi.
- Alleen te programmeren met Arduino IDE versie Arduino IDE 1.5.4

Communicatie tussen computer en Arduino ATmega32U4 via USB of Wifi via Arduino IDE (zie verderop).

Communicatie tussen ATmega32U4 en AR9331 via Arduino's Bridge Library (zie verderop).

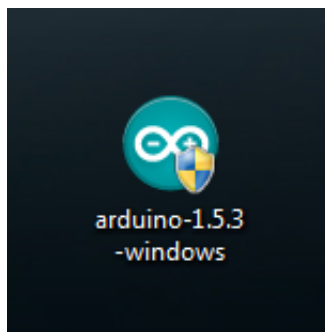
5. Drivers voor de Arduino Yun installeren

Op de Mac (OSX) hoef je niets te doen. Als je de Yun voor het eerst inpluigt verschijnt de "Keyboard Setup Assistant", deze kun je gewoon sluiten.

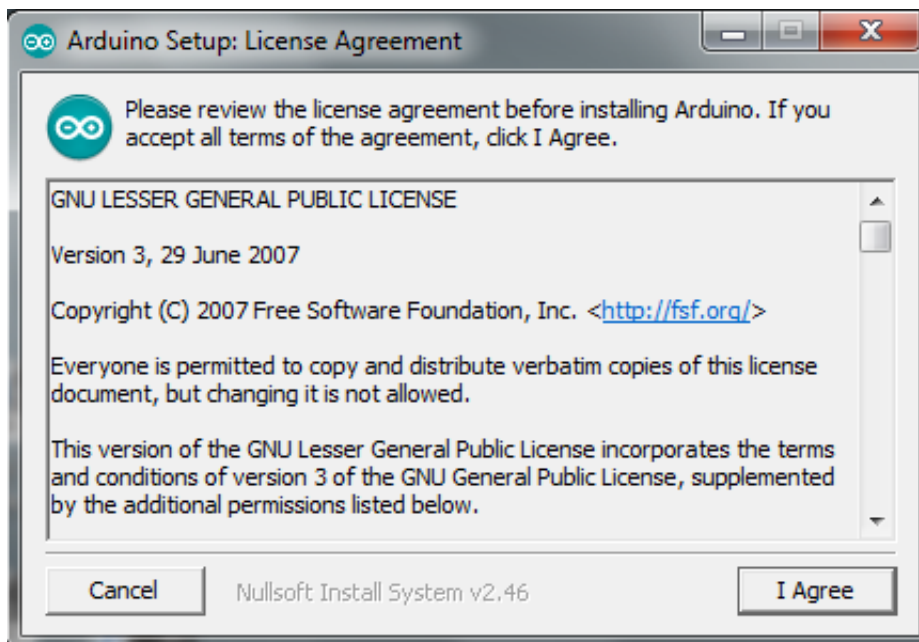


Windows

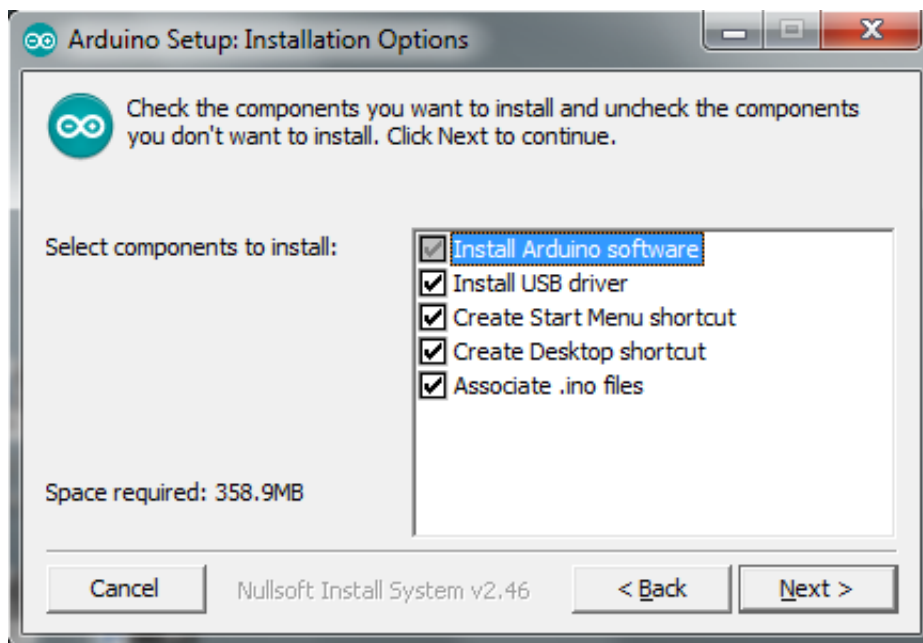
Voor windows moet je de Arduino IDE 1.5.4 installer downloaden. Dubbelklik op het installer icoon.



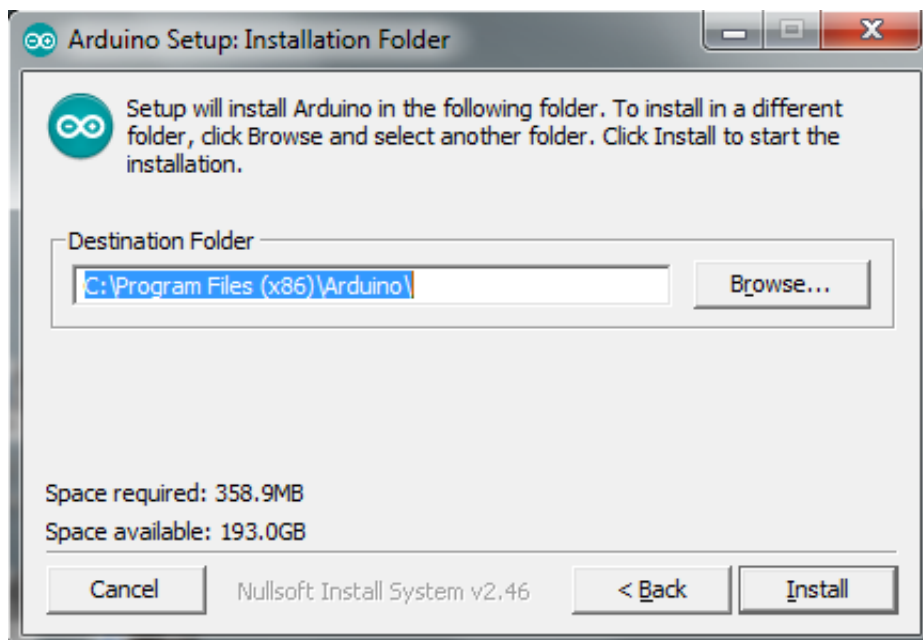
Klik op "I agree"



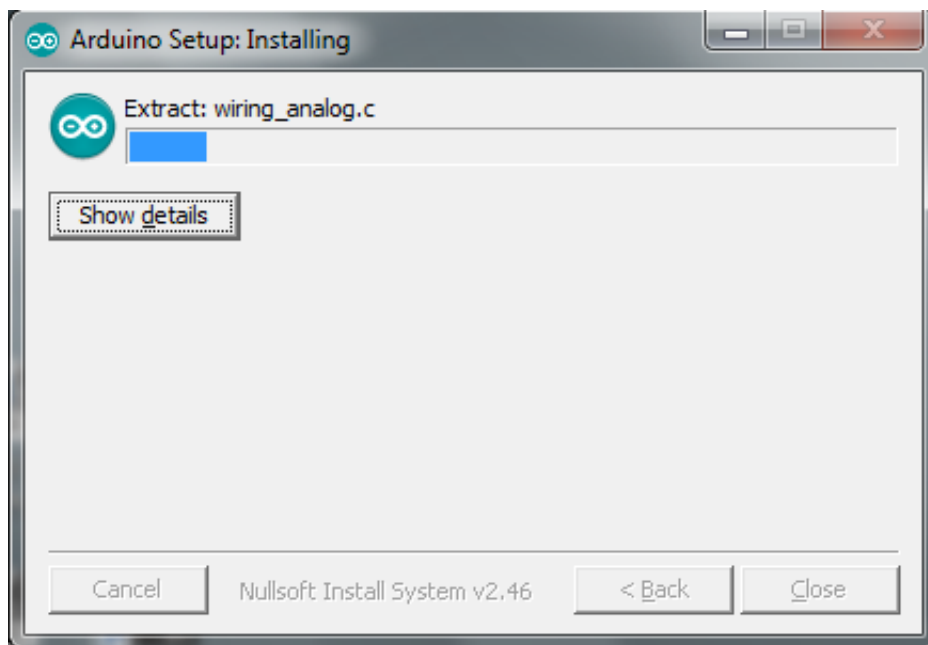
Klik op "Next".



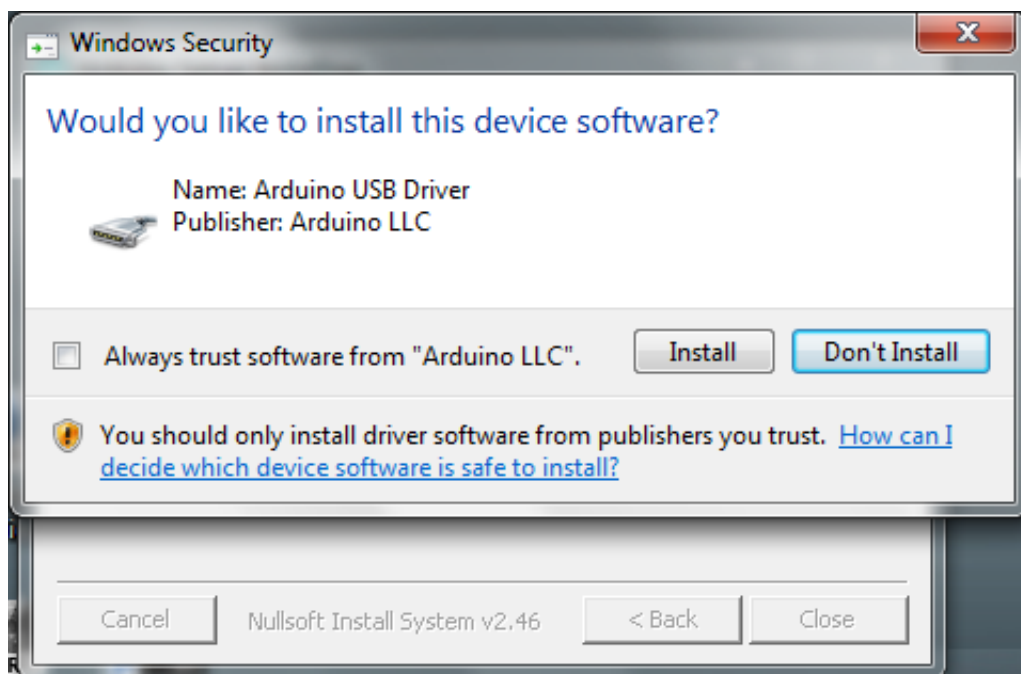
Selecteer de directory waar je de Arduino IDE geïnstalleerd wilt hebben en klik op "Install".



De installatie start.



Als je gekozen hebt voor het installeren van driver dan verschijnt onderstaande window. Klik "Install".



Als de installatie klaar is klik "Close".

Let op! De Yun gebruikt Bonjour services voor het auto-ontdekken van wireless networks. Deze service is niet standaard onderdeel van Windows. Als je Bonjour niet al geïnstalleerd hebt dan kun je het hier vinden en doen: [here](#). Aanvullend moet je erop letten dat anti-virus software of firewall software niet poort 5353 blocked voor communicatie.

Linux

Voor Ubuntu 10.0.4 en hoger hoeven er geen drivers geïnstalleerd te worden. Let wel op dat poort 5353 niet geblokkeerd wordt door een firewall.

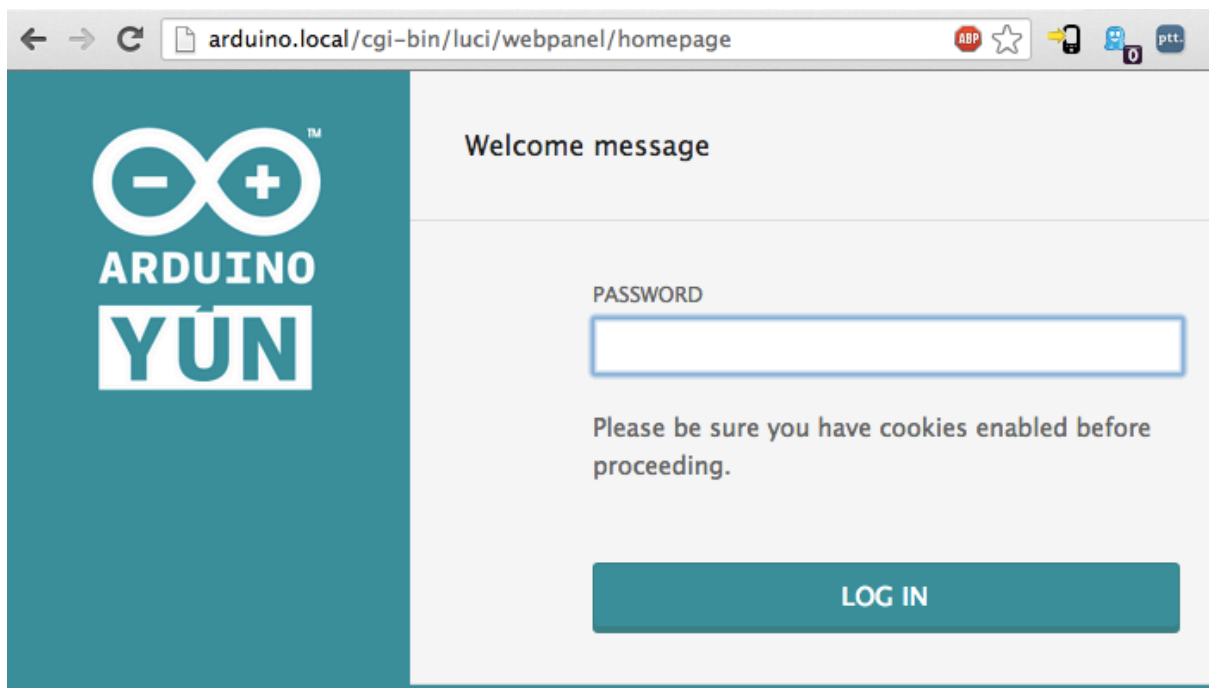
6. Onboard Wifi configureren

De Arduino Yun kan zowel als een Access Point gebruikt worden als verbinden met een bestaand netwerk. De onderstaande beschrijving verteld je hoe te verbinden met een bestaand netwerk. De Yun kan verbinden met unencrypted networks, of beveiligde zoals WEP, WPA en WPA2.

Als je de Yun voor het eerst aansluit zal het een Wifi netwerk creëren dat Arduino Yun-xxxxxetc. heet. Verbind jouw computer met dit netwerk.

Als je eenmaal verbonden bent met dit netwerk, open dan een browser en ga naar <http://arduino.local> of 192.168.240.1

Na een tijdje komt er een pagina te voorschijn die je vraagt naar een password. Type "arduino" in en klik op de "Log In" button.



Je komt daarna op een pagina terecht met diagnostische informatie over de huidige netwerk connecties. Het eerste overzicht betreft jouw Wifi interface, het tweede overzicht betreft jouw Ethernet connectie. Klik op "Configuration" om verder te gaan.

WELCOME TO ARDUINO, YOUR ARDUINO YÚN

CONFIGURE

WIFI (WLAN0) **CONNECTED**

Address	192.168.240.1
Netmask	255.255.255.0
MAC Address	B4:21:8A:00:00:10
Received	105.72 KB
Trasmitted	160.48 KB

WIRED ETHERNET (ETH1) **DISCONNECTED**

MAC Address	B4:21:8A:08:00:10
Received	0.00 B
Trasmitted	0.00 B

Op de nieuwe pagina zul je jouw Yun configureren door het een unieke naam te geven en door in te voeren aan welk netwerk je wilt koppelen of met welk netwerk je wilt verbinden.

In het Yun Name veld, geef jouw Yun een unieke naam. Deze zul je gebruiken om later naar de Yun te verwijzen.

Kies een password van 8 of meer karakters voor jouw Arduino. Als je het veld leeg laat blijft het default password 'arduino' het password.

Als je wilt kun je nu ook de timezone en het land invoeren. Het wordt aangeraden deze in te voeren omdat deze opties kunnen helpen bij het verbinden met locale Wifi netwerken.

Voer de naam in van het Wifi netwerk waarmee je wilt verbinden.

Kies het beveiligingstype van het netwerk en voer het password in.

YÚN BOARD CONFIGURATION

YÚN NAME *

MyYun

PASSWORD

••••••••

CONFIRM PASSWORD

••••••••

TIMEZONE *

America/New York

WIRELESS PARAMETERS

CONFIGURE A WIRELESS NETWORK ☒

WIRELESS NAME *

AccessPoint

SECURITY

WPA2

PASSWORD *

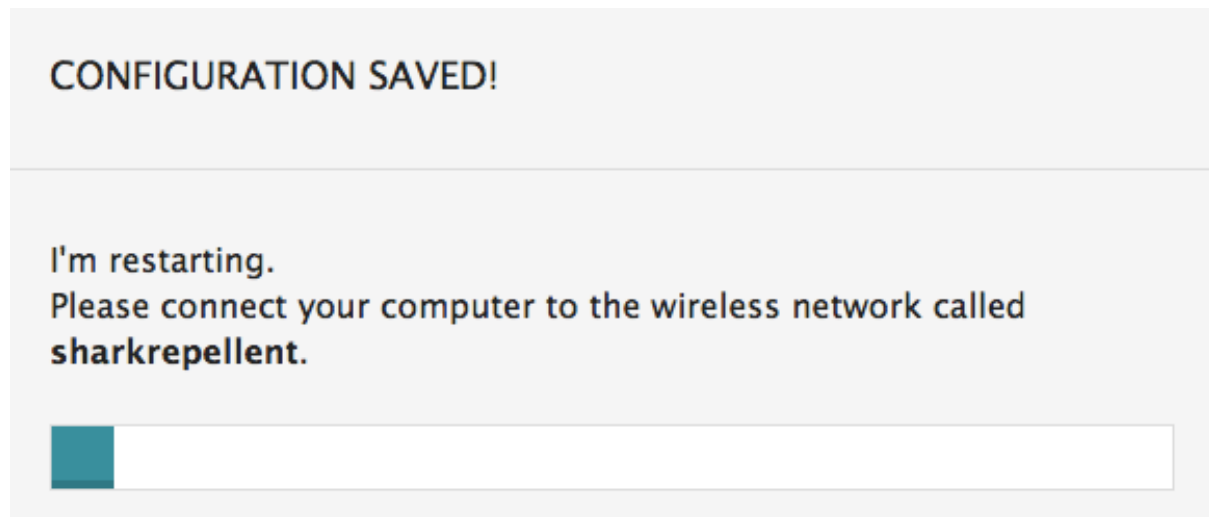
••••••••

DISCARD

CONFIGURE & RESTART

Als je op "Configure & Restart" klikt zal de Arduino Yun restarten en met het ingegeven netwerk verbinden. Het zelf gecreëerde arduino netwerk wordt opgeheven na een paar

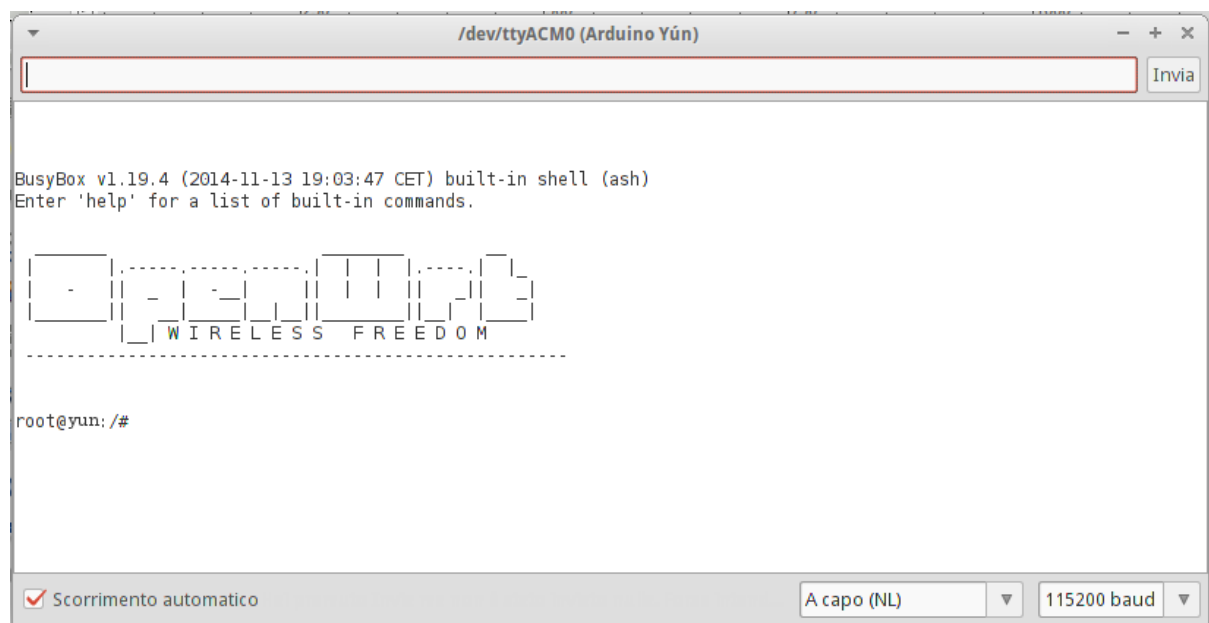
seconden.



Je kunt nu verbinden met het netwerk waarmee je de Arduino Yun hebt laten verbinden.

7. Reset het password van jouw Arduino Yun

Het wegpaneel password dat je zojuist hebt aangemaakt/veranderd kan opnieuw gereset of veranderd worden op de volgende manier. Verbind jouw Arduino Yun met een USB kabel aan de computer en upload de YunSerialTerminal sketch (onder Bridge examples in de Arduino IDE). Als het booten van de Yun klaar is, open de serial monitor, selecteer "New Line character" in het dropdown menu (onderin) en druk op Enter. Iets als het volgende zal verschijnen in de Serial Monitor.



Nu moet je het passwd command uitvoeren om het password te veranderen. Type **passwd** in de serial monitor and druk op enter. Je wordt gevraagd een nieuw password in

te voeren en deze nog een te herhalen. Als beiden passwords overeenkomen zal het systeem password gewijzigd worden en heb je toegang tot het wegpaneel met het nieuwe password.

8. Een virtuele serial port

Omdat de Yun geen toegewezen chip heeft om de serial communicatie af te handelen wordt er gebruik gemaakt van een 'virtuele' serial port. Dit is een software routine in het operating systeem en op het bord zelf. Net zoals de computer een seriële poort opent als je jouw Arduino inpluigt, creëert de 32U4 een seriële poort instantie als het de bootloader doorloopt. Dit betekend dat iedere keer dat je de 32U4 processor reset, de USB serial connectie verbroken wordt en opnieuw wordt opgebouwd/aangemaakt. Het bord (Yun) zal even uit de lijst van seriële poorten in de Arduino IDE verdwijnen en daarna weer verschijnen. Ieder programma dat op dat moment een connectie heeft open staan met de seriële connectie naar de Yun zal deze connectie verliezen. Dit loopt dus anders dan bij de Arduino Uno, waarbij een reset van de hoofdprocessor (de ATmega328P) niet resulteerde in het afsluiten van de USB connectie. Dit verschil heeft een aantal gevolgen voor driver installatie, uploaden en communicatie.

De Yun zal jouw sketch niet herstarten wanneer je een seriële poort opent op jouw computer. Dat betekend dat je geen serial data zult zien die al naar de computer gestuurd is door het bord, zoals bijvoorbeeld de data in de setup(). Dit geldt eveneens voor de Console als deze gebruikt wordt (Console wordt verderop uitgelegd). Dit betekend dat als je een Serial of Console print(), println() of write() statement gebruikt in de setup(), deze niet zichtbaar zal worden in de Serial monitor of Console connectie. Om hieromheen te werken kun je checken of de poort open is of niet, zie hieronder.

```
// stopt het programma totdat Serial Monitor geopend is
while (!Serial) ;
```

of

```
// stopt het programma totdat Console geopend is
while (!Console) ;
```

9. Programmeren met Arduino IDE via wireless connectie met Arduino Yun

Als de Yun op hetzelfde netwerk zit als jouw computer kun je er wireless mee verbinden om het te programmeren. Nadat je de Yun geconfigureerd hebt voor een wireless netwerk (zie hierboven) kun je de Arduino IDE openen (mits je op hetzelfde netwerk zit). Onder het Tools > Port menu moet je dan de Yun's naam en het IP adres zien. Selecteer onder het Board menu uiteraard de Arduino Yun.

Om een voorbeeld te draaien, kies het Blink voorbeeld (File > Examples > 01Basics > Blink) en upload deze sketch naar de Yun. Je zult gevraagd worden naar een administrator password, gebruik daarvoor het password dat je zelf aangemaakt hebt tijdens de eerdere configuratie. Als het programma is geupload zal de 32U4 processor herstarten en zul je de led die is aangesloten op pin 13 zien oplichten.

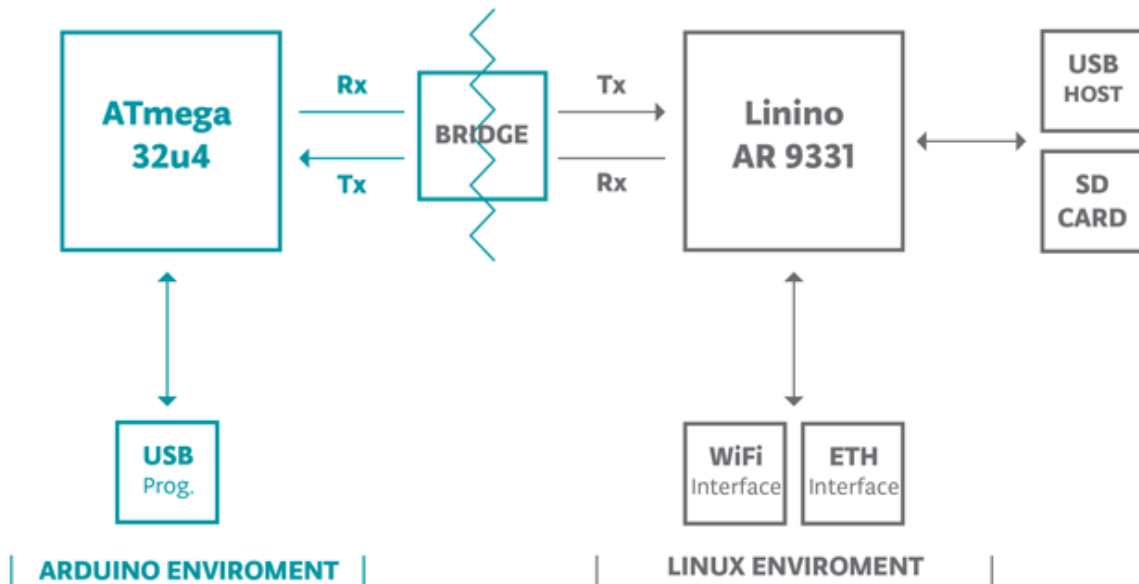
10. Programmeren met Arduino IDE via ethernet connectie met de Arduino Yun

Als je de Yun aansluit op een bedraad netwerk via een ethernet kabel, dan zal het proberen automatisch te verbinden via DHCP. Het bord verschijnt in het port menu net als via wifi connectie. Als je dit wilt doen, zorg dan dat de computer interface een statisch IP

adres heeft of als een DHCP server is ingesteld. NB: de ethernet interface is eth1, niet eth0.

11. Communicatie tussen Arduino en OpenWrt-Yun

De Bridge library maakt communicatie tussen de Arduino en OpenWRT-Yun mogelijk. Er zijn verschillende utility classes beschikbaar die verschillende vormen van communicatie tussen de twee mogelijk maken. Ze worden hieronder beschreven in meer detail. Voor een diepgaander overzicht zie [Bridge library reference](#) pagina's.



De wifi, ethernet interfaces, USB host en SD card zijn allemaal verbonden met de AR9331 processor. De Bridge library zorgt ervoor dat je toegang hebt tot deze devices, net als toegang tot het uitvoeren van scripts en communicatie met web services. Hieronder worden een aantal zaken uit de Bridge library besproken.

11.1 De Console

De Console, gebaseerd op de Bridge, maakt het mogelijk om informatie van de Yun naar een computer te sturen net als de serial monitor, maar dan wireless. De serial monitor met `Serial()` werkt niet als je niet via USB met de Yun verbonden bent. De Console zet een beveiligde verbinding op tussen de Yun en jouw computer via SSH (Secure Shell, zie verderop).

Laat het volgende op jouw Yun:

```
#include <Console.h>
```

```
const int ledPin = 13; // de pin waarop de LED is aangesloten
int incomingByte;      // een variabele om de binnenkomende serial data in te lezen
```

```
void setup() {
  // initialiseer serial communication:
  Bridge.begin();
}
```

```

Console.begin();

while (!Console){
  ; // wacht tot de Console port verbonden is
}
Console.println("You're connected to the Console!!!!");
// initialiseer de LED pin als een output:
pinMode(ledPin, OUTPUT);
}

void loop() {
  // kijk of er binnenkomende serial data is:
  if (Console.available() > 0) {
    // lees de oudste byte in de serial buffer:
    incomingByte = Console.read();
    // als het een hoofdletter H is (ASCII 72), zet de LED aan:
    if (incomingByte == 'H') {
      digitalWrite(ledPin, HIGH);
    }
    // als het een hoofdletter L is (ASCII 76) zet de LED uit:
    if (incomingByte == 'L') {
      digitalWrite(ledPin, LOW);
    }
  }
  delay(100);
}

```

Om de Console te zien, selecteer jouw Yun's naam en IP adres in het Port menu. De Yun staat alleen in het port menu (arduino IDE) als jouw computer en de Yun op hetzelfde LAN (netwerk) zitten. Open de port monitor en je wordt gevraagd naar de Yun's password.

Je kunt de Console ook zien door een terminal op de computer te openen en hierin te typen: `ssh root@yourYunsName.local 'telnet localhost 6571'` en dan enter.

NB: als je windows gebruikt moet je een terminal emulator installeren. PuTTY is een goede keuze. Je moet dan wel het ssh en het telnet commando apart invoeren.

Type nu H om de led op pin 13 aan te zetten en L om hem uit te zetten.

11.2 Process

Process commando's geven je de mogelijkheid om Linux processes op OpenWRT-Yun te runnen via jouw Arduino. Het volgende voorbeeld laat zien hoe je OpenWRT-Yun laat verbinden met een server met curl om ASCII tekst te downloaden en deze naar een serial connectie te printen (zie voorbeeld hieronder).

cURL is een tool die je helpt om data naar een server te sturen of van een server te plukken. Het ondersteunt een groot aantal protocollen zoals HTTP en FTP. Op de Yun is het waarschijnlijk dat je er HTTP requests mee doet om de inhoud van webpagina's op te vragen. cURL ondersteunt echter ook protocollen als FTP, POP3, IMAP en meer.

Om de inhoud van een website te lezen type je simpelweg `curl www.example.com` in en

op de Yun zal dan de HTML tekst van example.com beschikbaar komen. Voor meer informatie over cURL zie de cURL website: [structuring requests](#).

```
#include <Process.h>

void setup() {
  // Initialize Bridge
  Bridge.begin();

  // Initialize Serial, werkt niet als je via wifi met Arduino verbonden bent, dan moet je
  Console gebruiken ipv Serial.
  Serial.begin(9600);

  // Wait until a Serial Monitor is connected.
  while (!Serial);

  // run various example processes
  runCurl();
}

void loop() {
  // Do nothing here.
}

void runCurl() {
  // Launch "curl" command and get Arduino ascii art logo from the network
  // curl is command line program for transferring data using different internet protocols
  Process p;      // Create a process and call it "p"
  p.begin("curl"); // Process that launch the "curl" command
  p.addParameter("http://arduino.cc/asciilogo.txt"); // Add the URL parameter to "curl"
  p.run();        // Run the process and wait for its termination

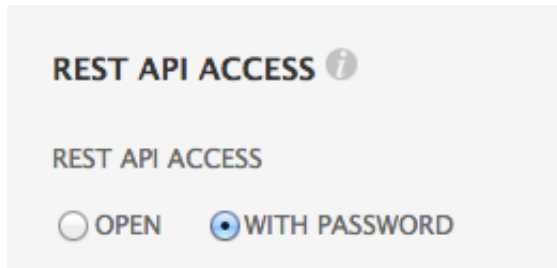
  // Print arduino logo over the Serial
  // A process output can be read with the stream methods
  while (p.available() > 0) {
    char c = p.read();
    Serial.print(c);
  }
  // Ensure the last bit of data is sent.
  Serial.flush();
}
```

11.3 Bridge gebruiken om informatie tussen de twee processoren uit te wisselen

Bridge maakt het mogelijk om informatie tussen de twee processoren uit te wisselen als key/value paren.

Dit voorbeeld laat zien hoe je de Bridge library kunt gebruiken om toegang te krijgen tot de digitale en analoge pinnen op het bord met behulp van REST calls. Het laat zien hoe je een API (application programmer interface) kunt maken die gebruikt maakt van REST calls door middel van een browser.

OperWRT-Yun gebruikt REST voor clients en servers. REST is een acroniem voor "Representational State Transfer". Het is een software architecture die delen van de Arduino hardware beschikbaar maakt voor de buitenwereld door middel van URLs. Default is de REST API van de Arduino beschermd met een password. Je kunt van de services gebruik maken als dit password wordt veranderd in de Yun configuratie panel. Onderaan de pagina zie je waar je de REST API ACCESS kunt wijzigen.



Een verdere introductie in REST en de achterliggende concepten is hier te vinden: [can be found here](#).

Er zijn REST eind punten herkenbaar voor de Yun:

- /arduino
- /data
- /mailbox

De /arduino directory heeft niet qua instellingen gepreconfigureerd. Alles wat aan de URL wordt toegevoegd wordt doorgegeven van de webserver naar de sketch op de 32U4. Je kunt de APIs in jouw sketch definiëren (zie een voorbeeld verderop in deze tekst: het Bridge voorbeeld).

"/data" wordt gebruikt om key/value paren op te slaan en in te lezen. De mogelijk calls zijn:

- */put/KEY/VALUE* : slaat een waarde op in de opslag
- */get/KEY* : vraag de waarde van de aangegeven key op in JSON
- */get* : haal de hele opgeslagen lijst op in JSON.
- */delete* : delete de interne opslag

"mailbox" wordt gebruikt om berichten aan een lijst toe te voegen die Mailbox genaamd wordt. De berichten lijst is opgeslagen in het Linux systeem en kan uitgelezen worden door de Arduino microcontroller. De beschikbare REST call is:

- *"/mailbox/message"*

Als je onderstaande voorbeeld draait, wees er zeker van dat jouw computer en de Yun op hetzelfde netwerk zitten.

Als je het bord geprogrammeerd hebt kun je de waarde van een pin opvragen, iets naar een pin schrijven en een pin als input of output configureren.

Het REST password moet wel uitgezet zijn, je kunt dan de volgende URLs gebruiken om zaken aan te sturen, op te vragen of te configureren:

- <http://myArduinoYun.local/arduino/digital/13> : calls digitalRead(13);
- <http://myArduinoYun.local/arduino/digital/13/1> : calls digitalWrite(13,1);
- <http://myArduinoYun.local/arduino/analog/9/123> : analogWrite(9,123);
- <http://myArduinoYun.local/arduino/analog/2> : analogRead(2);
- <http://myArduinoYun.local/arduino/mode/13/input> : pinMode(13, INPUT);
- <http://myArduinoYun.local/arduino/mode/13/output> : pinMode(13, OUTPUT);

Je kunt in plaats van URLs ook Curl commando's gebruiken vanaf een command line.

De volgende libraries moeten in jouw sketch opgenomen zijn:

```
#include <Bridge.h>
#include <YunServer.h>
#include <YunClient.h>
```

Initialiseer een server die het voor de Yun mogelijk maakt om naar verbonden clients te luisteren:

```
YunServer server;
```

Start in `setup()`, seriële (of console) communicatie voor debugging en zet LED op pin 13 hoog als de Bridge begint. `Bridge.begin()` blokkeert het programma, het duurt ongeveer 2 seconden. Als de Bridge gestart is, zet de LED uit.

```
void setup() {
  Serial.begin(9600); // of console wanneer je geen USB connectie hebt
  pinMode(13,OUTPUT);
  digitalWrite(13, LOW);
  Bridge.begin();
  digitalWrite(13, HIGH);
}
```

In het tweede deel van `setup()`, vertel je de instantie van de `YunServer` om naar inkomende verbindingen van de localhost te luisteren. Verbindingen gemaakt naar OpenWRT-Yun zullen doorgegeven worden aan de 32U4 processor om geparsed te worden en de pinnen aan te sturen. Dit gebeurt op poort 5555. Start de server met `server.begin()`.

```
server.listenOnLocalhost();
server.begin();
}
```

In de `loop()`, moet je nu een instantie van de `YunClient` maken voor het managen van de verbinding. Als de client verbinding zoekt/verbonden is, verwerk dan het commando in een speciaal daarvoor gemaakt functie (zie hieronder) en verbreek de verbinding wanneer het klaar is.

Een delay aan het einde van de `loop()` zorgt ervoor dat de processor niet teveel werk heeft.

```
void loop() {  
  YunClient client = server.accept();  
  
  if (client) {  
    process(client);  
    client.stop();  
  }  
  
  delay(50);  
}
```

Maak een functie die `process` heet en die de `YunClient` als argument accepteert. Lees het commando door een string te maken die de inkomende informatie opslaat. Parse het REST commando door de passende functionaliteit (digital, analog, of mode) en stuur de informatie door naar de juiste functie.

```
void process(YunClient client) {  
  String command = client.readStringUntil('/');  
  
  if (command == "digital") {  
    digitalCommand(client);  
  }  
  if (command == "analog") {  
    analogCommand(client);  
  }  
  if (command == "mode") {  
    modeCommand(client);  
  }  
}
```

Maak een functie die om kan gaan met de digital commando's. Accepteer de client als een argument. Maak locale variabelen om de pin en de waarde van het commando in op te slaan.

```
void digitalCommand(YunClient client) {  
  int pin, value;
```

Verwerk het client request voor de pin die aangegeven wordt met `client.parseInt()`.

Als het karakter na de pin een `"/"` is, betekend dat dat er een waarde van 1 of 0 volgt. Deze waarde wordt de waarde van de pin die de pin HIGH of LOW zet. Als er geen `"/"` is, dan moet de waarde van de pin gelezen worden.

```
pin = client.parseInt();  
  
if (client.read() == '/') {  
  value = client.parseInt();  
  digitalWrite(pin, value);
```

```

}
else {
  value = digitalRead(pin);
}

```

Print de waarde van de client en update de key/value waarden met de huidige waarde van de pin.

Door de waarde naar de client te schrijven met voorafgaand `F ()`, gebruik je het flash geheugen (memory). Dit helpt om de ruimte in SRAM zoveel mogelijk ongebruikt te laten, wat handig is bij het omgaan met lange strings zoals URLs.

De key is de pin en het type. Bijvoorbeeld D2 geeft de waarde voor digital pin 2. De waarde ervan zal de waarde zijn waar de pin op dat moment op gezet is, of wat er van de pin gelezen is.

```

client.print(F("Pin D"));
client.print(pin);
client.print(F(" set to "));
client.println(value);

String key = "D";
key += pin;
Bridge.put(key, String(value));
}

```

Om analog data te verwerken doen we hetzelfde, we noemen de key alleen A in plaats van D voor analoge input pins:

```

void analogCommand(YunClient client) {
  int pin, value;

  pin = client.parseInt();

  if (client.read() == '/') {
    value = client.parseInt();
    analogWrite(pin, value);

    // Send feedback to client
    client.print(F("Pin D"));
    client.print(pin);
    client.print(F(" set to analog "));
    client.println(value);

    String key = "D";
    key += pin;
    Bridge.put(key, String(value));
  }
  else {
    value = analogRead(pin);

```

```

client.print(F("Pin A"));
client.print(pin);
client.print(F(" reads analog "));
client.println(value);

String key = "A";
key += pin;
Bridge.put(key, String(value));
}
}

```

Nu nog 1 functie om pin mode veranderingen te managen. Accepteer wederom de YunClient als een argument in deze functie, maak een locale variabele om het pin nummer in op te slaan. Lees de pin waarde net als bij de digitale en analoge functie hierboven.

```

void modeCommand(YunClient client) {
  int pin;
  pin = client.parseInt();

```

Check of de URL klopt

```

if (client.read() != '/') {
  client.println(F("error"));
  return;
}

```

Als het een kloppende URL is, sla dan de URL op als string. Als de mode `input` of `output` is configureer de pin dan in de juiste mode en geeft terug aan de client dat het gedaan is. Als de string niet als waarde past geef dan een error message.
 String mode = client.readStringUntil('\r');

```

if (mode == "input") {
  pinMode(pin, INPUT);
  // Send feedback to client
  client.print(F("Pin D"));
  client.print(pin);
  client.print(F(" configured as INPUT!"));
  return;
}

if (mode == "output") {
  pinMode(pin, OUTPUT);
  // Send feedback to client
  client.print(F("Pin D"));
  client.print(pin);
  client.print(F(" configured as OUTPUT!"));
  return;
}

```

```

client.print(F("error: invalid mode "));
client.print(mode);

```

}

Je kunt nog veel meer voorbeelden vinden op de [Bridge Example](#) pagina of [Bridge library and examples](#) (nog verder uitwerken).

12. AR9331 met OpenWrt-Yun

12.1 Wireless en systeem configuratie

OpenWrt-Yun is gebaseerd op OpenWrt. De systeem configuratie kan middels command line maar ook via een website met configuratie settings via een interface die LuCi wordt genoemd. LuCi geeft toegang tot bijna alle settings om de Wifi interface aan te passen (je komt er door naar de localhost te gaan (Arduino web interface) en dan onderin het scherm vind je de optie om naar LuCi te gaan, of via Advanced, weet het niet meer zeker).

12.2 Installeren van software (packages)

Het installeren, updaten en verwijderen van additionele software op OpenWrt-Yun verloopt middels de Linux package manager opkg. In plaats van het moeten downloaden en compileren van bron code van het programma dat je wilt installeren is de bron (source) al gepreconfigureerd en gecompileerd volgens jouw specifieke systeem vereisten en in een pakket gestopt met andere informatie zoals versies en afhankelijkheden (vereisten van de software om het te kunnen installeren).

opkg maakt het mogelijk operaties op een package uit te voeren middels command line argumenten. De package manager (opkg) moet de database hebben die up to date is om alle packages die beschikbaar zijn voor jouw systeem te laten zien. —> het opkg update commando geeft een geüpdatete lijst van beschikbare packages. De database met lijst van packages wordt opgeslagen in RAM. Je moet daarom het opkg update commando iedere keer geven als je een programma wilt installeren nadat RAM is vrijgemaakt of nadat jouw systeem heeft gereboot.

```
root@Arduino:~#  
root@Arduino:~#  
root@Arduino:~# opkg update  
Downloading http://downloads.arduino.cc/openwrtyun/1/packages/Packages.gz.  
Updated list of available packages in /var/opkg-lists/attitude_adjustment.  
Downloading http://downloads.arduino.cc/openwrtyun/1/packages/Packages.sig.  
Signature check passed.  
root@Arduino:~#  
root@Arduino:~#  
root@Arduino:~# opkg install fortune-mod  
Installing fortune-mod (1.2.1-2) to root...  
Downloading http://downloads.arduino.cc/openwrtyun/1/packages/fortune-mod_1.2.1-2_ar71xx.ipk  
Configuring fortune-mod.  
root@Arduino:~#  
root@Arduino:~#  
root@Arduino:~# fortune  
Good news. Ten weeks from Friday will be a pretty good day.  
root@Arduino:~#  
root@Arduino:~#
```

Belangrijkste commands in opkg:

- update: update de package database (\$ opkg update)
- install: installeert een package, geef hierbij de naam mee (\$ opkg install fortune-mod)

- remove: verwijdert een geïnstalleerde package, geef hierbij de naam mee (\$ opkg remove fortune-mod)
- upgrade: upgrades een geïnstalleerde package naar een nieuwere versie, geef hierbij de naam mee (\$ opkg upgrade fortune-mod)
- list-installed: geeft lijst van geïnstalleerde packages op de Yun (\$ opkg list-installed)
- find: zoekt een package dat je wilt installeren (\$ opkg find fortune-mod)

Voor meer features van de opkg tool kijk: [reference on the OpenWrt website](#).

Om additionele software packages op de Yun te installeren (naast bijv. curl en python die standaard deel uitmaken van de installatie) moet het package manager systeem gebruikt worden (zie hierboven). Voor meer informatie hierover zie de [package manager tutorial](#) (nog verder uitwerken).

12.3 Communiceren met/tussen OpenWrt-Yun/Yun deel en computer

Voor de communicatie tussen jouw computer en de Yun (het Yun deel van de Arduino) wordt SSH gebruikt. SSH staat voor Secure Shell. Het is een terminal protocol om twee computers veilig met elkaar te verbinden.

Om te kunnen verbinden met de Yun vanaf de computer heb je het IP adres van de Yun nodig, het admin password en jouw computer en de Yun moeten op hetzelfde netwerk zitten.

Vind het IP adres van de Yun

Open de Arduino IDE, ga naar de 'ports' lijst en als je de Yun aangesloten hebt staat het adres hierin vermeld (onthoud of noteer dit).

Connect de computer met de Yun

Open nu een terminal applicatie op de computer en type het volgende in (achter de @ het ip adres van jouw Yun):

```
ssh root@172.16.1.190
```

Note: check als je de admin naam op de Arduino Yun verandert hebt of je dan niet ook de root als naam moet vervangen door deze nieuwe naam.

De computer probeert nu een connectie op te zetten met de Yun. De eerste keer dat je connect met jouw Yun op deze manier zul je het volgende zien verschijnen:

```
The authenticity of host '172.16.1.190 (172.16.1.190)' can't be
established. RSA key fingerprint is
d2:f2:e1:23:d7:29:dc:f1:68:48:58:99:ae:c3:64:6f. Are you sure you want
to continue connecting (yes/no)?
```

Type yes dan enter en ga verder en je krijgt het antwoord: Warning: Permanently added '172.16.1.190' (RSA) to the list of known hosts.

Daarna wordt je gevraagd een password in te voeren. Gebruik hiervoor het password dat je hebt bedacht bij het configureren van jouw Arduino Yun.

De SSH connectie is nu gemaakt en vanaf nu kun je via de terminal communiceren met de Yun en OpenWRT via een Command Line Interpreter (CLI).

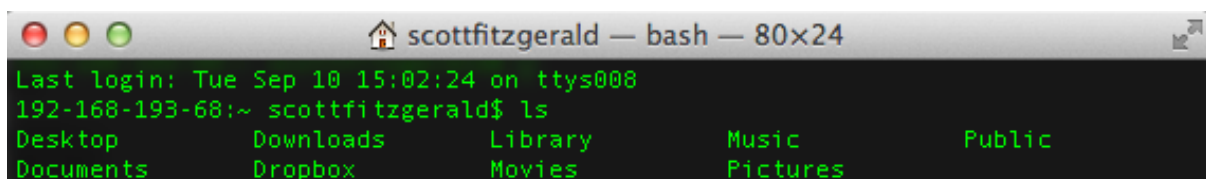
12.4 CLI

Communiceren met OpenWrt-Yun kan middels een command line, ofwel via de Arduino's Bridge Library, ofwel via SSH. Commands worden verstuurd door een command line interpreter (CLI). Zo kun je commando's sturen die applicaties runnen, configureren van het operating systeem, die diagnostische informatie laten zien, of wat je maar zou willen doen. Voor meer informatie over de basics van de Command line zie: [the Debian documentation](#).

Met een terminal emulator op een computer kun je een connectie openen tussen jouw Yun en jouw computer, zodat je informatie kunt versturen en kunt ontvangen en applicaties kunt runnen van de Yun's Linux distributie. Om dit te doen start je een terminal (bij windows heb je PuTTY nodig). Enige voorbeelden van een command ingeven met CLI zijn:

- ls (geeft een lijst van items in de huidige directory) (zie plaatje hieronder)
- ls -R (geeft lijst van items in de huidige en alle onderliggende/sub directories) (zie plaatje hieronder)
- mkdir (maak een directory, geef de naam van de directory daarbij mee).
- cd (naam van de directory, verander de huidige directory naar de nieuwe)
- cd .. (ga 1 directory terug/hoger)
- chmod (verander de file permissies van een file)
- cp (kopieer een item)

Een lijst van veel gebruikte Linux commands en hun flags (extra commands die je mee wilt geven aan een command) is hier te vinden: [Debian help site](#). Het volgende document heeft meer informatie over het gebruik van CLI en OpenWRT: [using the CLI and OpenWRT](#).



```
scottfitzgerald — bash — 80x24
Last login: Tue Sep 10 15:02:24 on ttys008
192-168-193-68:~ scottfitzgerald$ ls
Desktop      Downloads    Library      Music        Public
Documents    Dropbox     Movies       Pictures
```

```
Arduino — bash — 80x23
192-168-193-68:Arduino scottfitzgerald$ ls -R
libraries

./libraries:
DmxMaster      TKLCD          TinkerKit      readme.txt

./libraries/DmxMaster:
DmxMaster.cpp  DmxMaster.h    README.md       examples        keywords.txt

./libraries/DmxMaster/examples:
FadeUp         SerialToDmx

./libraries/DmxMaster/examples/FadeUp:
FadeUp.ino

./libraries/DmxMaster/examples/SerialToDmx:
SerialToDmx.ino

./libraries/TKLCD:
INSTALL.txt    TKLCD.cpp      examples
LICENSE.txt    TKLCD.h        matrix_lcd_commands.h

./libraries/TKLCD/examples:
```

12.5 Upgrade de OpenWRT versie op jouw Yun

Het kan zijn dat de versie van jouw OpenWRT-Yun achterhaald is. Bekijk de [upgrade tutorial](#) om jezelf ervan te verzekeren dat je de nieuwste versie van de software gebruikt (dit later nog uitbreiden met de instructies om de OpenWRT te upgraden).

12.6 Externe opslag op de Yun (microSD card)

Je wordt ontmoedigd het interne non-volatile geheugen van de Yun te gebruiken omdat het maar een beperkt aantal writes toestaat. Je kunt gebruik maken van een externe microSD kaart of ander extern medium om jouw data, scripts, web pagina, etc. op te bewaren. De Yun kan deze informatie bereiken en er informatie opslaan door de 32U4 processor als er in de root van de opslag een directory staat die `arduino` heet.

13. Python

Een installatie van Python 2.7 is standaard onderdeel van OpenWRT-Yun, waarmee je applicaties of scripts kunt schrijven voor het Linux gedeelte van de Arduino Yun. Meer over Python: [Python 2.7 documentation pages](#).

Als je net begint met Python zijn er een aantal hele goede resources: "[Learn Python the Hard Way](#)" verteld je alles wat je moet weten over Python en het schrijven van jouw eigen scripts.

14. Acknowledgement

The text of the Arduino getting started guide is licensed under a [Creative Commons Attribution-ShareAlike 3.0 License](#). Code samples in the guide are released into the public domain.