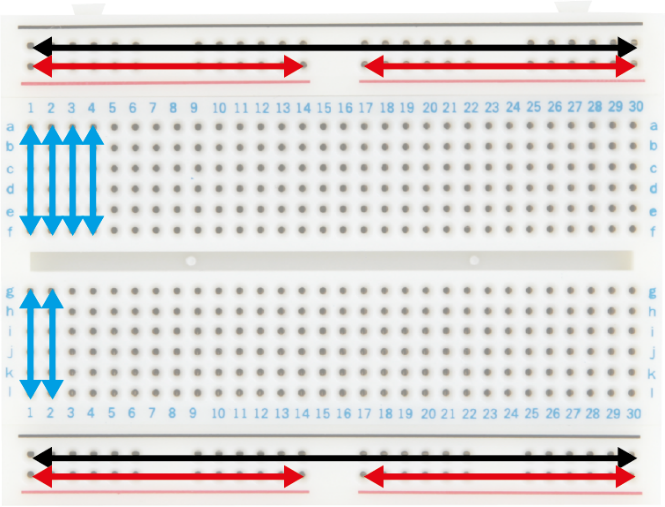
MVP1: Pure Elektronica

In deze fase gaan we een LockDownLamp maken op basis van pure elektronica, de Lamp is vrij simpel in wat het doet: druk op de knop en het lichtje gaat aan, laat de knop los en het lichtje gaat uit. Iedereen in het gezin krijgt een drukknop en een lichtje, om aan te geven dat ze niet gestoord mogen worden.

# Wat heb je nodig?

* Een breadboard
* Een ESP32 (een microcontroller)
* Male-male kabeltjes
* Per gezinslid
  + Een LED
  + Een drukknop
  + Een weerstand van 100Ω (bruin-zwart-bruin-goud) (uitleg op <https://opencircuit.nl/Kleurcodering> )

# Breadboard

Om dit te maken gaan we een breadboard gebruiken, zodat we niet hoeven te solderen en we de onderdelen achteraf ook anders kunnen gebruiken.

Een breadboard is een plaatje waarbij de gaatjes gebruikt worden om elektronica in te steken en met elkaar te verbinden. De gaatjes zijn op een speciale manier verbonden: de buitenste rijen zijn horizontaal verbonden (zwart voor min en rood voor plus) en de kolommen daartussen zijn verticaal verbonden (blauwe pijlen).

Let wel dat het Velleman breadboardje een speciale is en een breuk heeft in de plus lijnen, dus ga je de rode delen wel nog onderling moeten gaan verbinden.

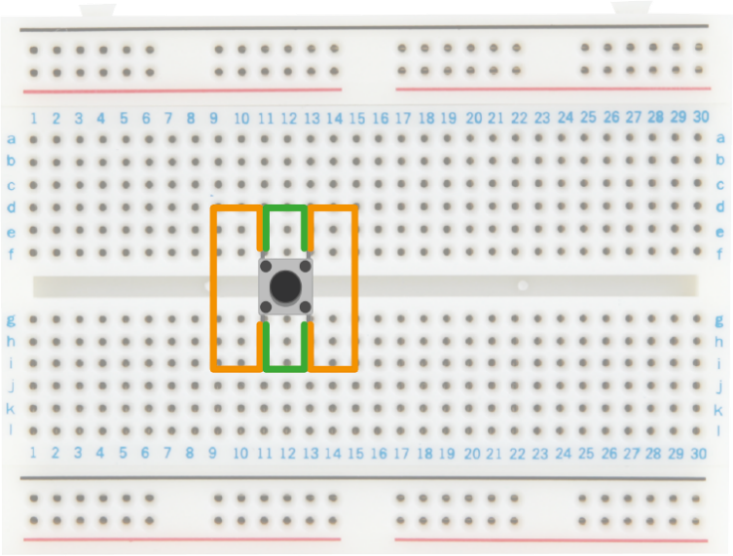
Naast de breadboard heb je twee grote dingen nodig: een LEDje en een drukknop. Ik ga ze even kort uitleggen.

# Het LEDje

Het LEDje in een lampje dat licht geeft als er stroom door loopt van plus naar min: een LEDje heeft een plus (anode) en een min (kathode) kant, dus als je het LEDje omgekeerd aansluit doet het niets. Hoe weet je wat de plus kant is? Een LEDje heeft twee pootjes en het langste pootje is de plus. Maar zelfs als je de pootjes hebt afgeknipt kan je de plus kant van de LED voelen aan de rand op de LED, die is plat aan de min kant en steekt uit aan de plus kant… Dus de stroom moet van de rode rij naar de lange poot van de LED en dan via de korte poot naar de zwarte rij….

MAAR als er te VEEL stroom door het LEDje gaat dan gaat het LEDje kapot, dus moet je er best een weerstand tussen steken die ervoor zorgt dat er nooit teveel stroom door het LEDje gaat… voor deze opstelling is een weerstand van 100Ω genoeg, die herken je aan de bruin-zwart-bruin-gouden bandjes op de weerstand! Hoe groter de weerstand hoe minder fel het LEDje gaat branden.

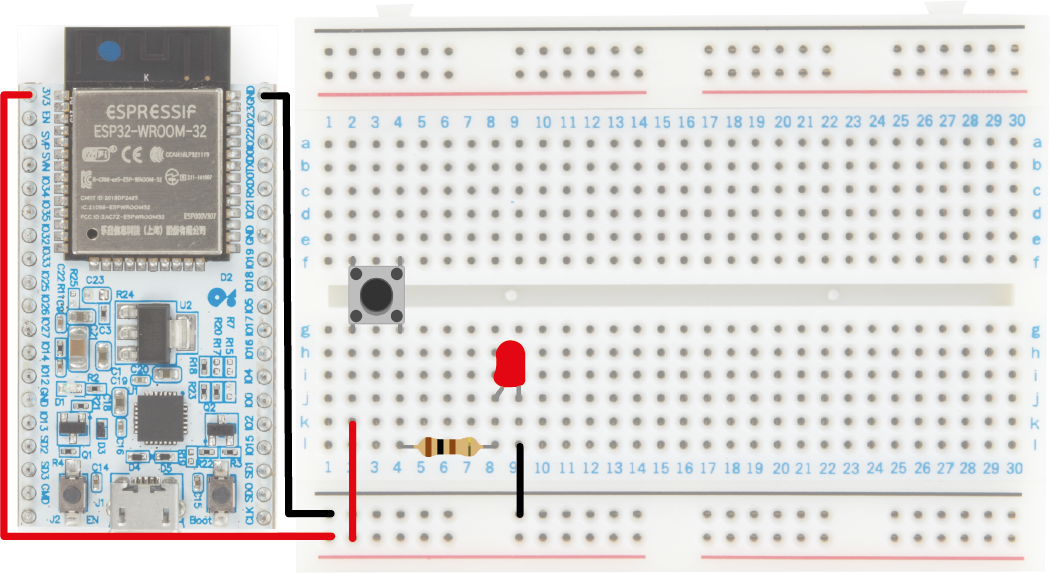
# AB12-E130 フラットコマンドスイッチ AB12 1個 富士電機 【通販サイト ...De Drukknop

De drukknop is simpel: de knop verbindt de twee kanten van de drukknop zodra je de drukknop indrukt. De pootjes die met de orange lijnen zijn verbonden staan ALTIJD verbonden (of je de drukknop indrukt of niet). De pootjes die met de groene lijnen zijn verbonden, verbinden alleen als de drukknop is ingedrukt. Je kan dit ook onthouden als de pootjes die aan dezelfde kant van de knop staan, zijn alleen verbonden als je de drukknop indrukt. Let op dit is een veelgemaakte fout.

# Stroomkring

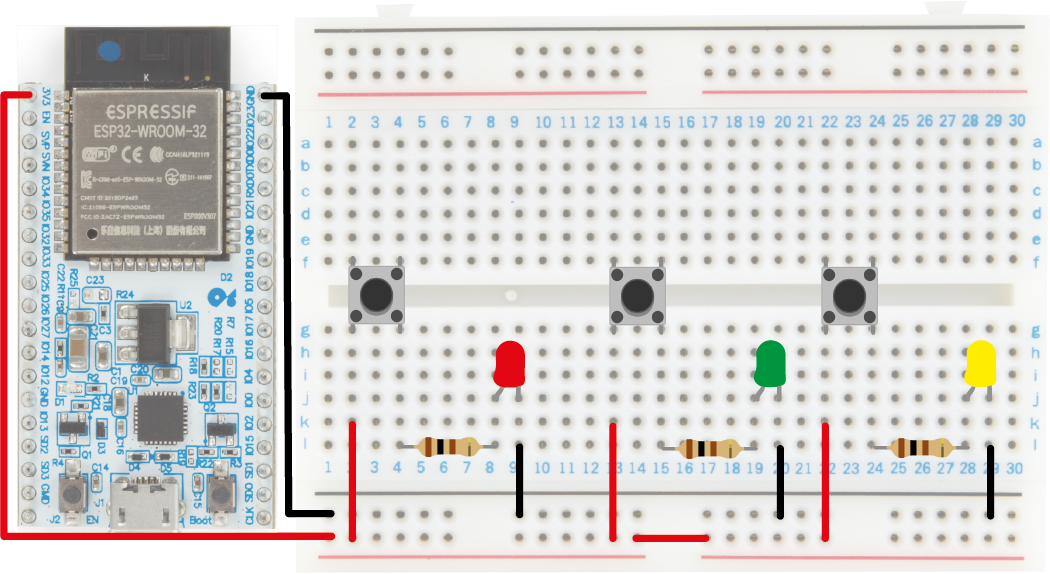
In de eerste lamp gaan we een stroom kring maken, waarbij de stroom van plus naar min loopt over de drukknop en de weerstand en het LEDje (dat licht geeft). De stroom kan je bekomen van een stroombron (batterij), maar wij gaan hiervoor de ESP32 gebruiken. Als je de rode lijn connecteert aan de 3.3V pin en de zwart lijn aan de GND pin, dan heb je een rij met plus (rood) en een rij met min (zwart). Nu moet de stroom van + naar de drukknop gaan en van daar naar de weerstand en de LED en dan naar min. Dat doe je voor elke set van drukknop en LED, voor elk lid van je gezin/kot.

Maar laten we beginnen met 1 lamp en 1 knop om de basis te begrijpen:



Als we op de breadboard naar de hoek links onder kijken dan gaat op kolom 2 een draadje van plus naar de drukknop en een draadje van het korte pootje van de LED in kolom 9 naar de min. Tussen de lange poot van de LED in kolom 8 en een andere poot van de drukknop in kolom 4 zetten we een weerstand van 100Ω (bruin-zwart-bruin-goud)… Als je nu de ESP32 via een USB kabel verbindt met je computer of een GSM oplader, dan zal er stroom door de ESP32 gaan (er gaat een LEDje op de ESP32 oplichten). Als je dan alles goed hebt gedaan gaat het LEDje oplichten als je op de drukknop blijft drukken.

Als alles werkt kan je nu de andere twee knoppen en LEDjes doen:



Gebruik de afbeelding om de knoppen en LEDjes correct te connecteren en vergeet niet dat de plus lijn op deze breadboards is onderbroken en je de linker en de rechter kant met elkaar gaat moeten verbinden. Let wel dat je op de breadboard veel meer knoppen en LEDjes kan zetten, want de knoppen hoeven niet in het midden te staan. Je kan alles ook veel dichter op elkaar zetten omdat de weerstanden niet 4 gaten hoeven in te nemen, maar hier doen we dit voor de overzichtelijkheid.

Je kan zelf de vorm en kleur van de drukknop kiezen en ook de kleur van het LEDje. Dus wees gerust creatief!

Ook is het wel lastig dat je aldoor maar op de drukknop moet blijven drukken om het LEDje aan te houden… Je kan dit oplossen door er iets op te leggen met een gewicht of door te kijken naar een andere soort van schakelaar: een wasknijper met twee stukken metaalfolie kan ook een schakelaar zijn zolang het de stroomkring kan onderbreken…

MVP2: Een slimmere LockDownLamp

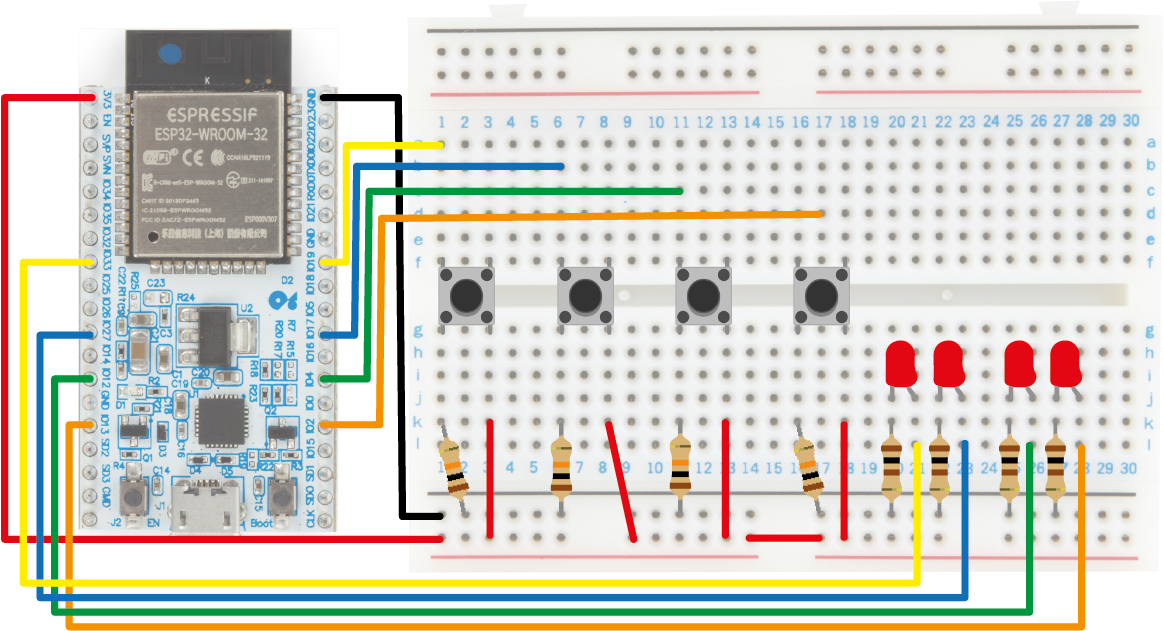
In deze fase gaan we de ESP32 gebruiken als microcontroller. De drukknoppen gaan een input signaal sturen naar de ESP32 en die gaat op zijn beurt een output signaal sturen naar de LEDjes. Eigenlijk niet veel anders dan de vorige opstelling, maar het heeft een aantal grote voordelen:

* Je hoeft de drukknop niet meer constant ingedrukt te houden
* Je kan de LockDownLamp slimmer maken in de programmatie, zo kan slechts een van de ouders op niet storen staan etc.

# Wat heb je nodig?

* Een breadboard
* Een ESP32
* Male-male kabeltjes
* Per gezinslid
  + Een LED
  + Een drukknop
  + Een weerstand van 100Ω (bruin-zwart-bruin-goud)
  + Een weerstand van 10kΩ (bruin-zwart-orange-goud)

# Schema:



# LEDs

In principe is er aan de LEDjes niet veel verandert, er zit nog steeds een weerstand achter om ervoor te zorgen dat er niet te veel stroom door de LEDjes gaat. Maar het grote verschil is dat ze nu niet aan de plus hangen naar elk een plekje hebben gekregen op de ESP32 (pinnen 13, 12, 27 en 33). Let op de kleuren van de kabels in het schema, de kleuren komen overeen met de kleuren van de kabels naar de drukknoppen zodat de ziet welke drukknop aan welke LED hangt, maar in principe is de input van de drukknop aan de overkant van de output van de LED op de ESP32. In de code zijn de inputs en outputs gelinkt in deze groepen:

|  |  |  |
| --- | --- | --- |
| Groep | LEDPIN (output) | ButtonPIN (input) |
| 1 | 13 | 2 |
| 2 | 12 | 4 |
| 3 | 27 | 17 |
| 4 | 33 | 19 |

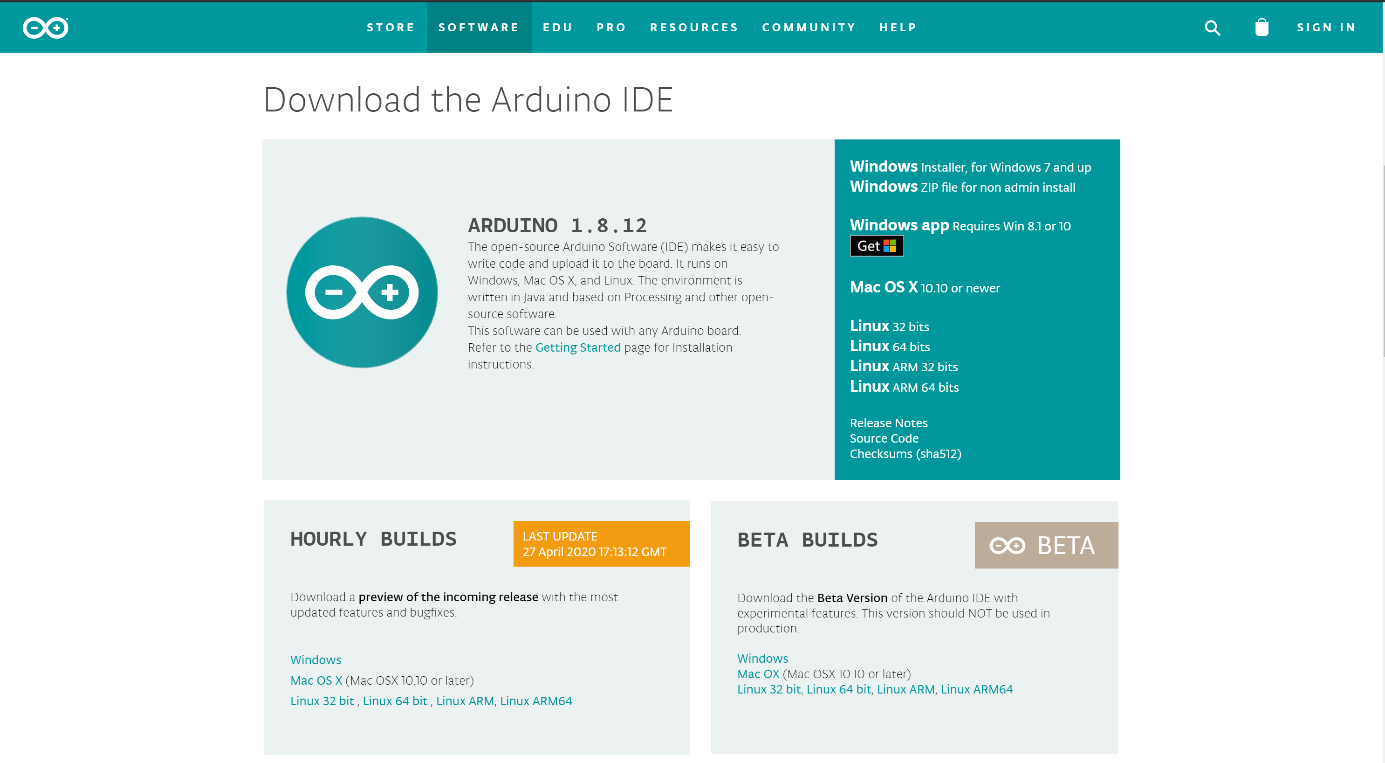
# Drukknop

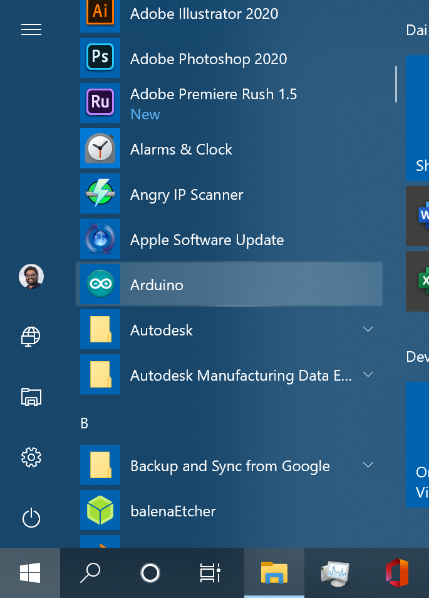
De drukknoppen zijn hier ook wat specialer, je voegt een weerstand toe om ervoor te zorgen dat er een duidelijke hoog en laag stand is voor de input pin. Als er niets verbonden is met de input pin dan betekent dat niet dat de pin op nul staat, de ESP32 gaat dan zelf beslissen wat het denkt dat het is en dan krijg je rare gedragingen van de input: de ene keer denkt het dat het aan staat en de andere dat het uit staat. Dus een weerstand zorgt ervoor dat er een duidelijke hoog en laag is voor de input pin.

# ESP32

Nu kan je de hele opstelling maken op basis van het schema hierboven, maar de knoppen en LEDjes gaan NIETS doen, omdat ze moeten worden aangestuurd door de ESP32 en die moet je nog programmeren. Haal je computer nu boven en sluit de ESP32 aan op je computer met een USB kabel.

Ga naar: <https://www.arduino.cc/en/main/software> en scroll naar beneden tot je dit ziet en kies de juiste versie voor je computer. Download de app en installeer dit.

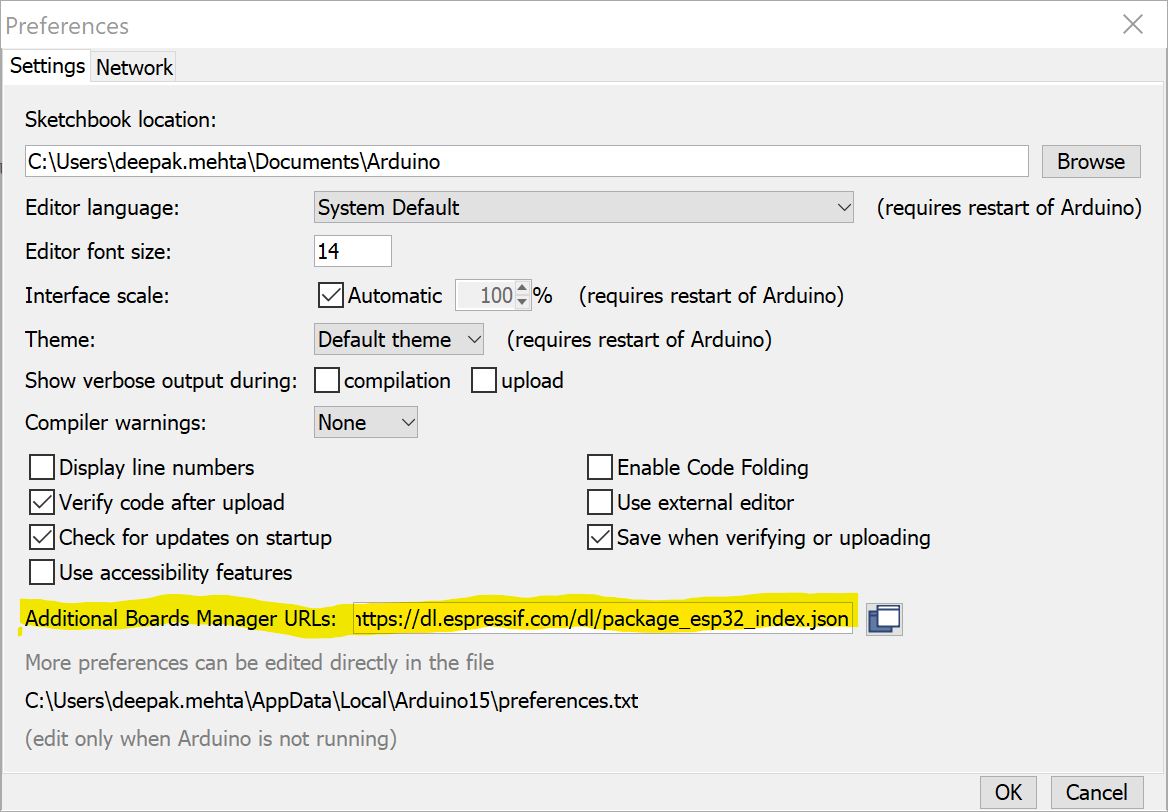
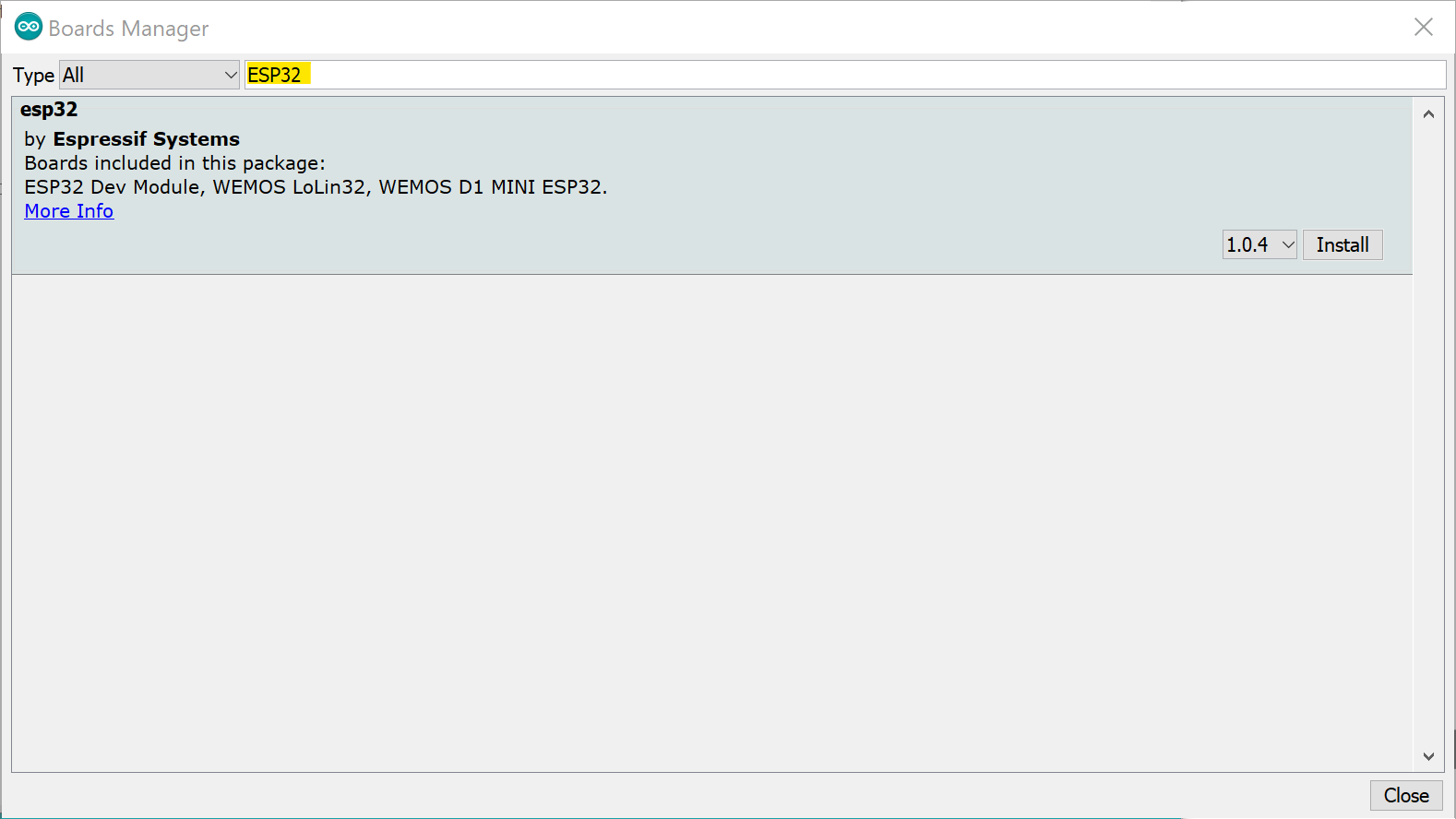
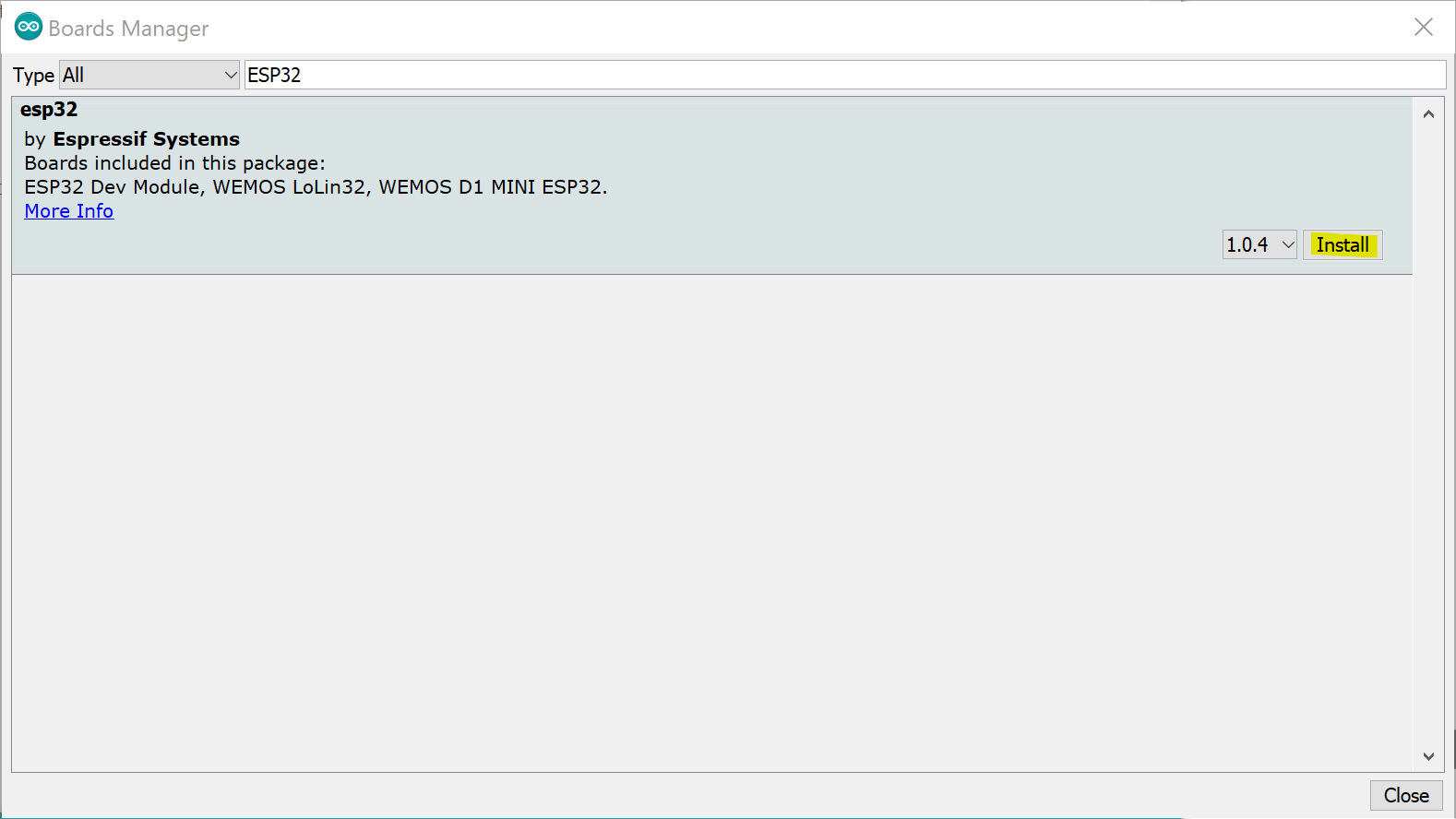


Je gaat nu een Arduino app vinden. In Windows is dat in je Start menu:

Als je de Arduino app opstart dan kom je op een venster als deze:



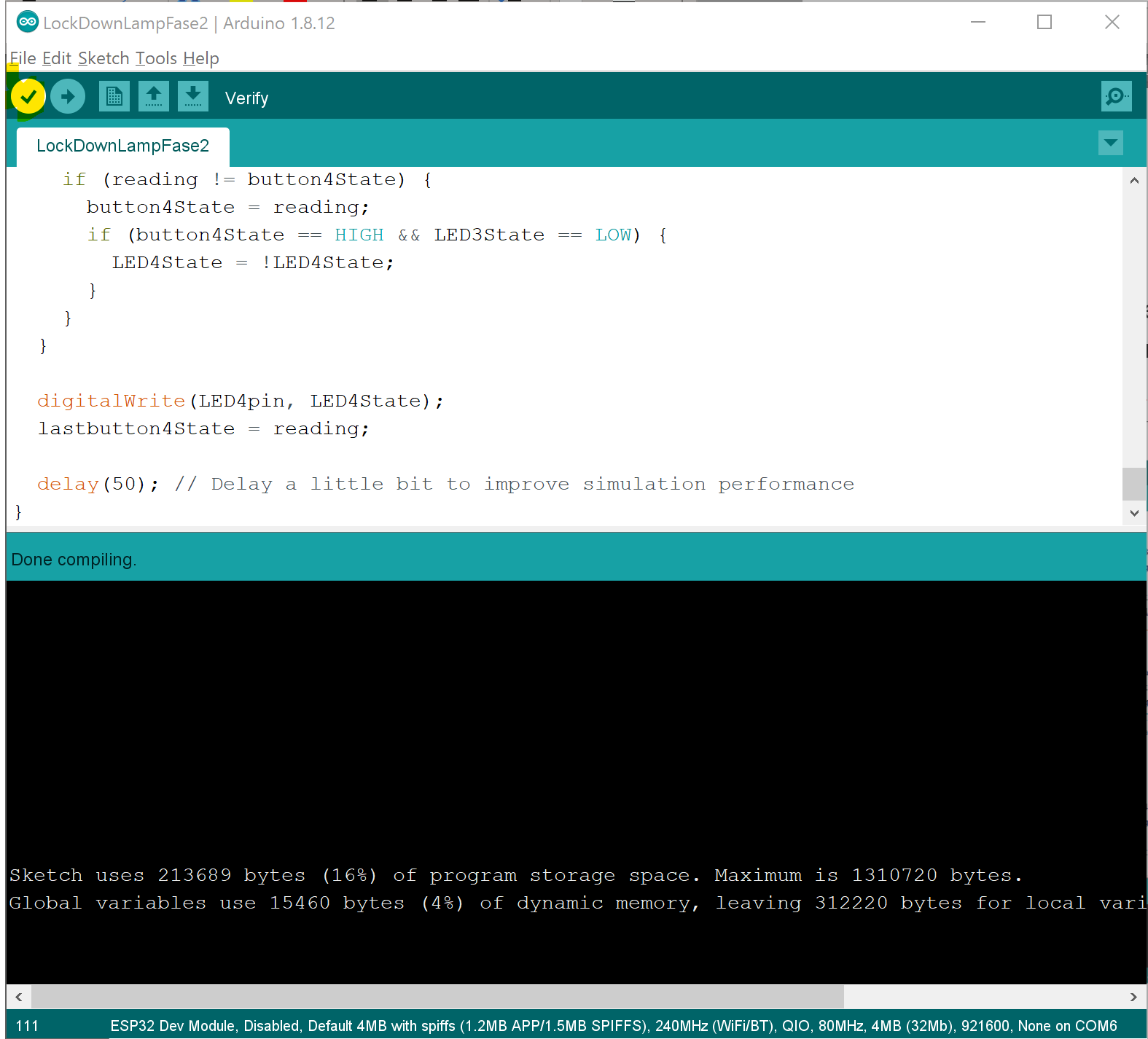
We gaan nu eerst zorgen dat de Arduino app onze ESP32 herkent en begrijpt, daarvoor ga je de ESP32 borden moeten installeren. Twee stappen:

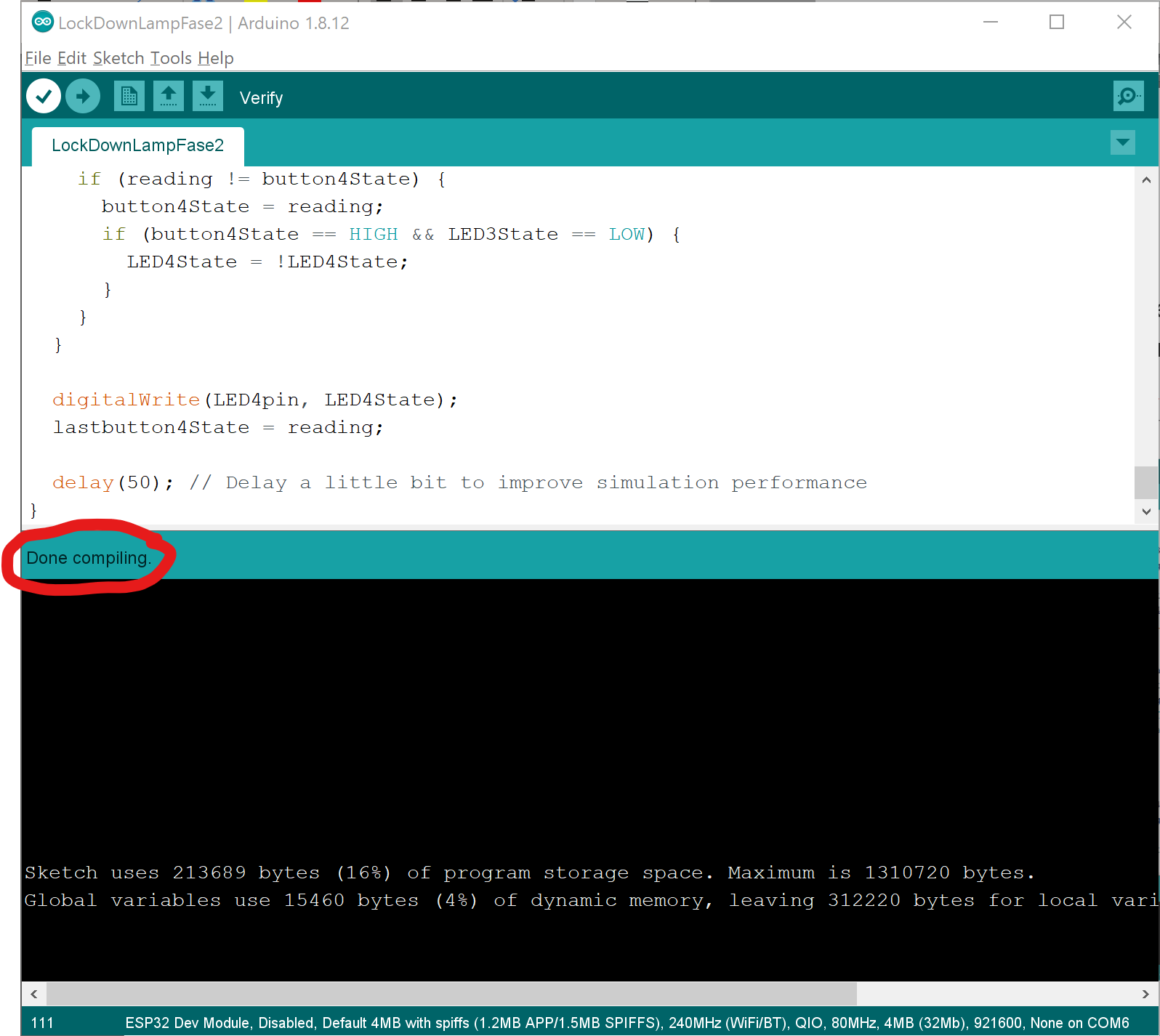
1. Ga naar File/Preferences of Bestand/Voorkeuren en in de Additional Boards Manager URLs zet je de volgende URL: <https://dl.espressif.com/dl/package_esp32_index.json> en druk dan op OK.  
    
2. Ga dan naar Tools/Board:/Boards Manager… en typ dan ESP32 in de tekst balk boven:  en klik dan op install. 

Nu ben je klaar om je ESP32 te programmeren met de Arduino App. Onder de Tools/Boards: menu zal je een hele boel ESP32 kaartjes zien staan, wij moeten de “ESP32 Dev Module” hebben. Als je de ESP32 nu verbindt met de computer zal je ook de juiste COM poort kunnen selecteren en dan ben je klaar om je ESP32 te programmeren!

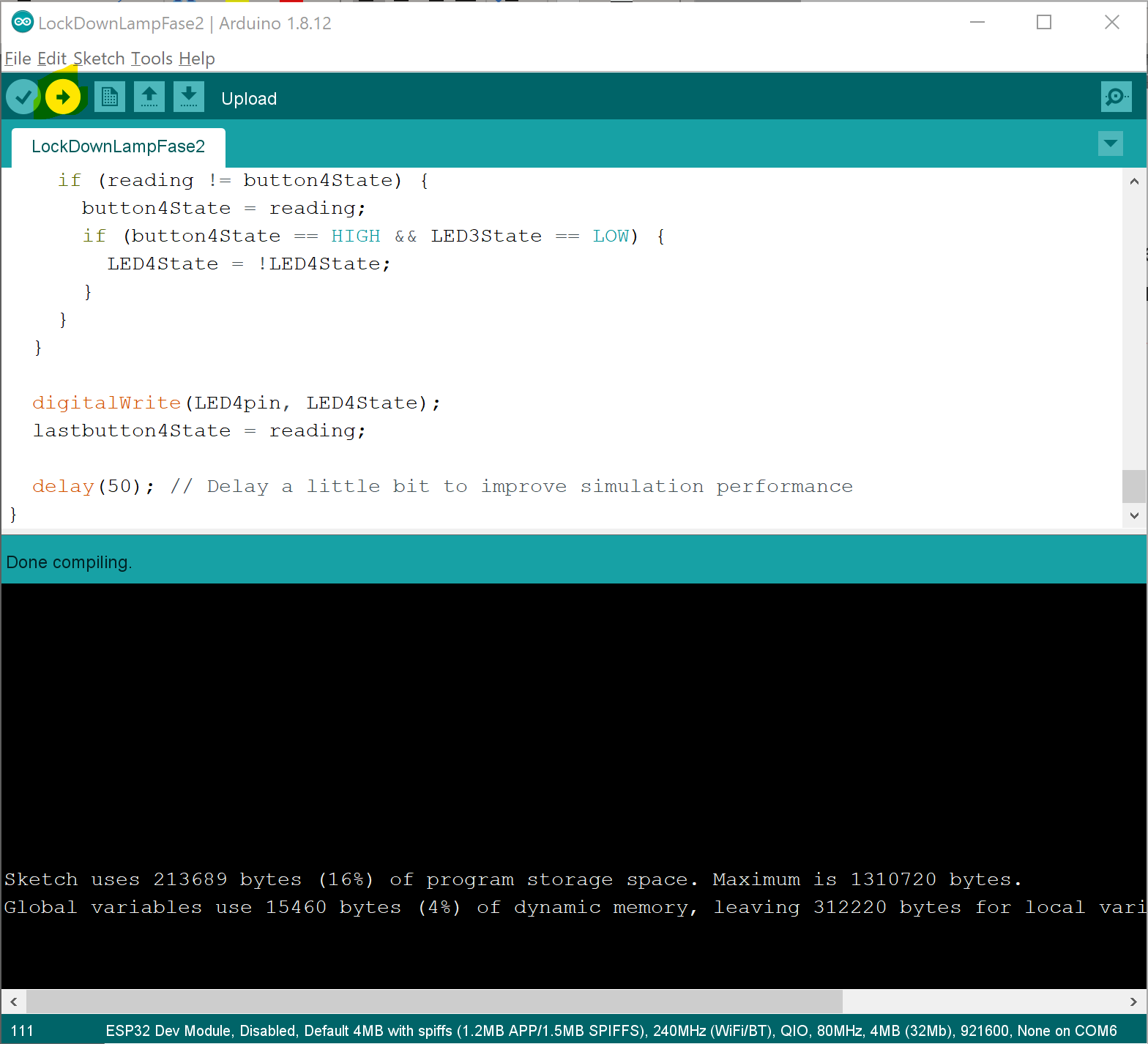
Om het allemaal wat makkelijker te maken, hebben we het programma al gemaakt, zodat elke knop aan een LED is gekoppeld, niets slims, maar gewoon een keer drukken op de knop om de LED aan te zetten en nog eens drukken om de LED af te zetten. Je hoeft de LED niet meer ingedrukt te houden. Hoe gaan we het programma van de computer in de ESP32 laden?

Link Programma Fase 2: 

Eerst open je het programma in de Arduino app: Ga naar File/open… en open het .ino bestand. De app zal je vragen om een nieuwe folder te maken, doe dat zeker. Connecteer de ESP32 aan je computer met een USB kabel en selecteer de juiste Board (ESP32 Dev Module) en de juiste COM poort.

Druk op de Verify knop om het programma te controleren op fouten… Normaal zullen er geen fouten in zitten.

Na een tijdje zal je de boodschap krijgen “Done Compiling” en dan weet je dat alles OK is.

Nu kan je op de knop naast Verify drukken, die Upload heet… Eerst gaat de app nu weer het verify proces doen , maar daarna gaat het het programma op de ESP32 laden. Aan het einde van het proces zie je wat tekst in het zwarte stuk van het venster verschijnen met als laatste tekst:

Leaving...

Hard resetting via RTS pin...

Dan weet je dat de ESP32 werkt op het programma dat je net hebt opgeladen…. En nu kan je testen! Deze versie zorgt dat alle knoppen een LEDje schakelen, MAAR LEDje 4 kan alleen aan als LEDje 3 niet aan staat… dit is een kleine aanzet naar hoe je de knoppen slimmer kan maken.

Gefeliciteerd je hebt je LockDownLamp nu slimmer gemaakt en kan hem programmeren zoals je die wil. Je kan ook een teller inbouwen zodat je nooit meer dan 3 keer per dag niet kan worden gestoord of dat niet meer dan 2 LEDs per keer aan kunnen staan of dat bv. Beide ouders niet samen kunnen gaan werken… Je hebt het programma en kan nu op verkenning gaan om het te proberen aan te passen naar wat jij wil dat de LockDownLamp doet…

MVP3: De Internet LockDownLamp

De ESP32 heeft ook Bluetooth en Wifi… dus waarom gooien we de knoppen niet allemaal weg? Fysieke drukknoppen is zo vorige eeuw! Klikbare links op webpaginas zijn de nieuwe drukknoppen! Wat als de ESP32 zijn eigen webserver draait en een pagina heeft waarop je de LEDjes direct kan aansturen! Dan hoef je je kamer zelfs niet meer uit komen om iedereen te laten weten, dat je niet gestoord wil worden!

# Wat heb je nodig?

* Een breadboard
* Een ESP32
* Male-male kabeltjes
* Per gezinslid
  + Een LED
  + Een weerstand van 100Ω (bruin-zwart-bruin-goud)

# Schema:

A circuit board

Description automatically generated

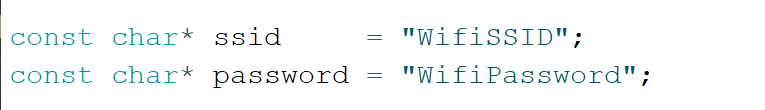
We hebben gewoon alle drukknoppen weg gehaald van de vorige opstelling, je kan de LEDjes wat meer vrijheid geven nu, maar wij laten het zo…

Het enige wat er verandert met de vorige stappen is dat je een andere .ino file gaat uploaden naar je ESP32…

Link Programma Fase 3:

Je gaat de file openen in de Arduino app en dan moet je TWEE dingen aanpassen!

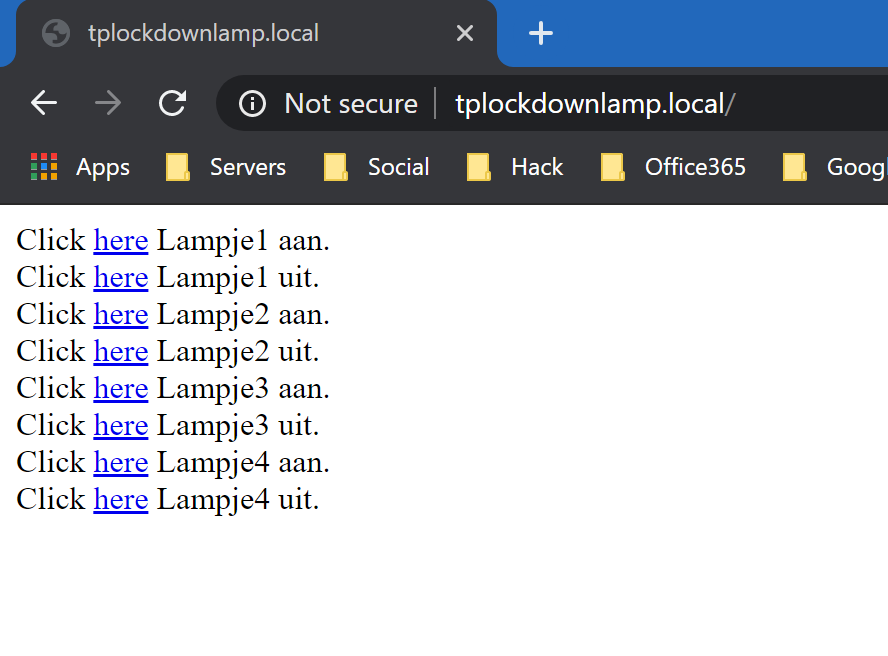
Eerst op regel 34 & 35 vul je de juiste Wifi gegevens in:





Deze zorgen dat je ESP32 kan connecteren met je Wifi netwerk.

Op regel 75 kan je ook de naam van je ESP32 veranderen, wij noemen hem tplockdownlamp, maar je kan hem gerust een andere naam geven… Deze naam ga je gebruiken om met je webbrowser de ESP32 op je wifi netwerk te vinden… je typt <http://tplockdownlamp.local> en je browser connecteert met je ESP32.

Maar eerst bewaar je de file en dan verify en upload… Als je nu op je smartphone of eenderwelke browser op je Wifi netwerk naar <http://tplockdownlamp.local> gaat zal je een mooi kleine webpagina krijgen waar je elke LED aan en uit kan zetten!

Als je goed kijkt naar de code of de pagina, zie je dat je elke lamp kan controleren door de juiste URL naar de ESP32 te sturen via de webbrowser!

<http://tplockdownlamp.local/H1> zet lamp 1 aan

<http://tplockdownlamp.local/L1> zet lamp 1 uit

Als je deze URLs in je bookmarks steekt kan je met een druk op de knop vanuit je browser je LEDje aanzetten en zal niemand je nog storen…. Zalig!

Natuurlijk gelden hier ook de slimme dingen die je in Fase 2 al kon doen.

En nu heb je een volledig slimme internet-geconnecteerde LockDownLamp gemaakt in 3 weken. Geef je lamp de nodige extras:

* Versier de Lamp zodat het niet op een hoopje elektronica lijkt
* Gebruik andere outputs dan een LEDje: met een klein servo motortje kan je ook hele leuke dingen doen.
* De ESP32 kan ook gegevens van het internet halen, zo kan je het weer betrekken.
* Maak de Lamp ook deel van een echt IoT netwerk zodat de Lamp dat ze je mogen storen als er eten wordt opgewarmd.