

Treball final de grau

Estudi: Grau en Enginyeria Electrònica Industrial i Automàtica

Títol: Pantalla RGB dinàmica gestionada de manera remota

Document: 1. Memòria

Alumne: Wei Li

Tutor: Albert Figueras Coma

Departament: Enginyeria Elèctrica, Electrònica i Automàtica

Àrea: Enginyeria de Sistemes i Automàtica

Convocatòria (mes/any) Juny/2019

ÍNDEX

1. INTRODUCCIÓ	3
1.1. Antecedents.....	3
1.2. Objecte	3
1.3. Abast	3
2. LA MATRIU DE LEDS	4
2.1. Tipus de díodes LEDs SMD.....	4
2.2. LEDs WS2812B.....	5
2.2.1. Característiques WS2812B.....	5
2.2.2. Funcionament de WS2812B	6
2.2.3. Matriu 8X32 LED WS2812B flexible	8
2.3. Programacions de les matrius LEDs	8
2.3.1. Introducció de NeoPixels	8
2.3.2. La llibreria de NeoPixels	9
2.3.3. La llibreria de NeoMatrix.....	10
2.3.4. La llibreria Adafruit_GFX.....	10
2.4. Verificar i dissenyar el cablejat de fabricació de matrius	11
2.4.1. Matriu simple	12
2.4.2. Matriu complexa	13
2.4.3. Combinació total de matrius	15
3. NODEMCU EL KIT DE DESEVOLUPAMENT	16
3.1. La selecció de MCU amb seu kit de desenvolupament.....	16
3.2. La introducció del SoC ESP8266.....	18
3.3. Mòduls ESP-12E	19
3.4. Kit o placa de desenvolupament NodeMCU.....	21
3.4.1. Versions de NodeMCU	21
3.4.2. Accés als pins de NodeMCU V3.....	22
3.4.3. Pins digitals de NodeMCU V3.....	23
3.4.4. Pin analògic de NodeMCU V3	23
3.4.5. Pins d'alimentació de NodeMCU V3	24
3.4.6. LEDs i polsadors de NodeMCU V3.....	25
4. PREPARACIÓ DEL ARDUINO IDE PER ESP8266	27
4.1. La instal·lació del controlador	27
4.2. La instal·lació d'Arduino IDE en sistemes Windows	27
4.3. Afegir suport al IDE per ESP8266.....	29

5. PROGRAMANT EL ESP8266 PER VIA SENSE FIL.....	32
5.1. Instruccions de l'OTA.....	32
5.2. Seguretat bàsica de l'OTA	33
5.3. Requisits bàsics de l'OTA	33
5.3.1. Las instal·lacions	33
5.3.2. Càrrega a través d'un port sèrie.....	34
5.3.3. Troba el port OTA.....	35
5.3.4. Recàrrega nova amb OTA.....	36
6. FONTS D'ALIMENTACIÓ PER LA PANTALLA LEDS RGB.....	38
6.1. Fonts d'alimentació.....	38
6.2. Sistema de protecció elèctrica	39
6.3. Adaptador de nivell	39
6.3.1. Introducció Level shifer	39
6.3.2. Esquema de muntatge.....	40
7.MANUAL DE PROCEDIMENTS DE MARXA D'EQUIP.....	42
7.1. Muntatge físic del sistema	42
7.2. Connexió amb port serial	42
7.3. Actualització amb OTA	43
8. RESUM DEL PRESSUPOST.....	44
9. CONCLUSIONS	45
10. RELACIÓ DE DOCUMENTS	46
11. BIBLIOGRAFIA.....	47
12. GLOSSARI	48
A. CÀLCUL DE PROTECCIÓ	49
B. PROGRACIÓ.....	50

1. INTRODUCCIÓ

Els avenços de la tecnologia electrònica, han aconseguit un ràpid desenvolupament de les comunicacions. Fent que una matriu de LEDs RGB sigui comandat per un circuit de control basat en un microcontrolador, d'una manera molt senzilla i eficaç.

1.1. Antecedents

Actualment en molts comerços publiciten els seus productes i donen informació d'ofertes, esdeveniments, etc. als seus clients en les pissarres tradicionals, per exemple: els partits de futbol, els menús, les tapes, etc. Això és molt incòmode, han de escriure-ho i esborrar-ho sovint, no son aplicables ni bonics.

1.2. Objecte

Per tant, l'objectiu del projecte serà actualitzar les informacions a una pantalla de LEDs RGB dinàmica que puguin visualitzar-les amb diferents colors. La informació que es visualitzarà s'haurà de poder escriure i enviar de manera remota amb un mòbil o portàtil PC.

1.3. Abast

Es muntarà una pantalla de dimensió 128cmx16cm que conté 8 mòduls de tipus WS2812b (matriu 32x8 LEDs RGB) per mostrar les lletres i els números contínuament. Controlarà la pantalla un kit NodeMCU que consta d'un xip microcontrolador ESP8266, connectat a través d'un BUS de comunicació sèrie. El mòbil o el PC portàtil es connectarà sense fils amb el NodeMCU per enviar-li la informació a mostrar per pantalla.

2. LA MATRIU DE LEDS

Quan va a triar un tipus d'il·luminació LED la majoria de les vegades es basa en aspectes estètics o de lluminositat. Llavors, primer s'introduiran en els tipus de díodes LEDs. Veuran quines característiques tenen cada un i en què es diferencien.

2.1. Tipus de díodes LEDs SMD

Aquests díodes LEDs són els més estesos del mercat. Són utilitzats tant per a il·luminació domèstica com professional. El díode ve encapsulat en una resina semirígida, de manera que assegura una bona protecció enfront de cops. Proporciona una gran quantitat de llum unidireccional.

Es poden trobar gran varietat al mercat aquest tipus de LED. Els més comuns són 3528 i 5050.

3528: Se'ls coneix per aquest nom a causa de les mesures d'encapsulat, que són 3,5mmx 2,8mm. Són bastants fiables i s'utilitzen per bombetes petites i mitjanes amb baixa potència. Permet el control, són uns dels més econòmics del mercat. Es col·loquen per exemple en les tires de LED.

5050: La mida 5mmx5mm. Són més fiables i tenen millor rendiment que els anteriors. A causa de la seva alta potència són un dels més usats del mercat. En tenir 3 nuclis (cada díode està dividit en una part color vermell, una altra blava i una altra verda) són utilitzats en les tires o matrius de LEDs RGB.



Figura 1. Circuit de un LED RGB

2.2. LEDs WS2812B

Els WS2811, WS2812 i WS2812B són LEDs estan basats en el LED 5050, que disposen de lògica integrada, per la qual cosa és possible variar el color de cada LED de manera individual (a diferència de les tires RGB convencionals en què tots els LEDs canvien de color de forma simultània).

La genial novetat del WS2812B (i resta de família) és afegir un integrat dins de cada LED, que permet accedir a cada píxel de forma individual. Per aquest motiu aquest tipus de LED es denominen "Individuals Adreçables".

Això obre la porta a una infinitat d'aplicacions i combinacions, que van des dotar d'il·luminacions diferents zones amb una única tira, animacions complexes, o fins i tot generar pantalles senceres d'alta lluminositat.

La figura 2 mostra diverses tipus de WS2812B:

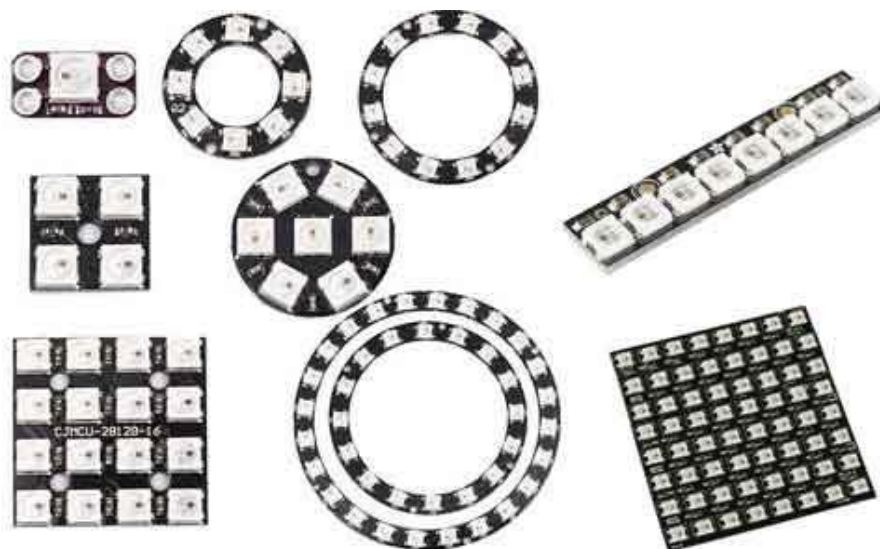


Figura 2. Tipus de LEDs WS2812B

2.2.1. Característiques WS2812B

Protecció de connexió inversa intel·ligent, la connexió inversa de la font d'alimentació no danya l'IC.

És un LED de baix consum i alt brillantor, que incorpora en un únic encapsulat els 3 colors RGB. El circuit de control i el LED comparteixen l'única font d'alimentació. El xip RGB estan integrats en un paquet de 5050 components, que formen un control complet de píxels de punt.

El circuit de remodelació del senyal incorporada, després de la remodelació de l'ona al següent controlador, garanteix la distorsió de la forma d'ona no acumular. Circuit de restabliment elèctric incorporat i circuit de restabliment de pèrdua d'energia.

Cada píxel dels tres colors primaris pot arribar a una pantalla de 256 brillantors, completat 16777216 a tot color a la pantalla, i freqüència d'escaneig no menor a 400Hz / s. Senyal de transmissió de port en cascada per una sola línia. Cada dos punts assenyalen la distància a més de 5 m del senyal de transmissió sense cap circuit d'augment.

2.2.2. Funcionament de WS2812B

Tal com s'ha l'integrat WS2812B emmagatzema 3 bytes (24 bits), la raó és perquè utilitza 1 byte per cada microled / píxel de color RGB. Amb un byte (o 8 bits) es pot emmagatzemar valors de 0 a 255, això significa que cada microled / píxel RGB pot tenir fins a 256 nivells.

Com es pot dir-li a cada LED que tingui un color determinat i totes les dades passen per sol cable? La solució és senzilla: quan un LED rep un flux de bytes, emmagatzema els 3 últims bytes rebuts i transmet la resta al següent LEDs que farà el mateixa. Finalment, amb un senyal que es diu "resetcode" cada LED mostra l'últim valor que té emmagatzemat.

Es imaginem que tenim una tira de 5 LED si es volen encendre el LED1 de vermell, el LED3 de color verd i el LED5 de color blau, deixant la resta apagats. Doncs bé, la idea és llançar un enfilall de bits d'aquesta manera que demostren a la figura següent:

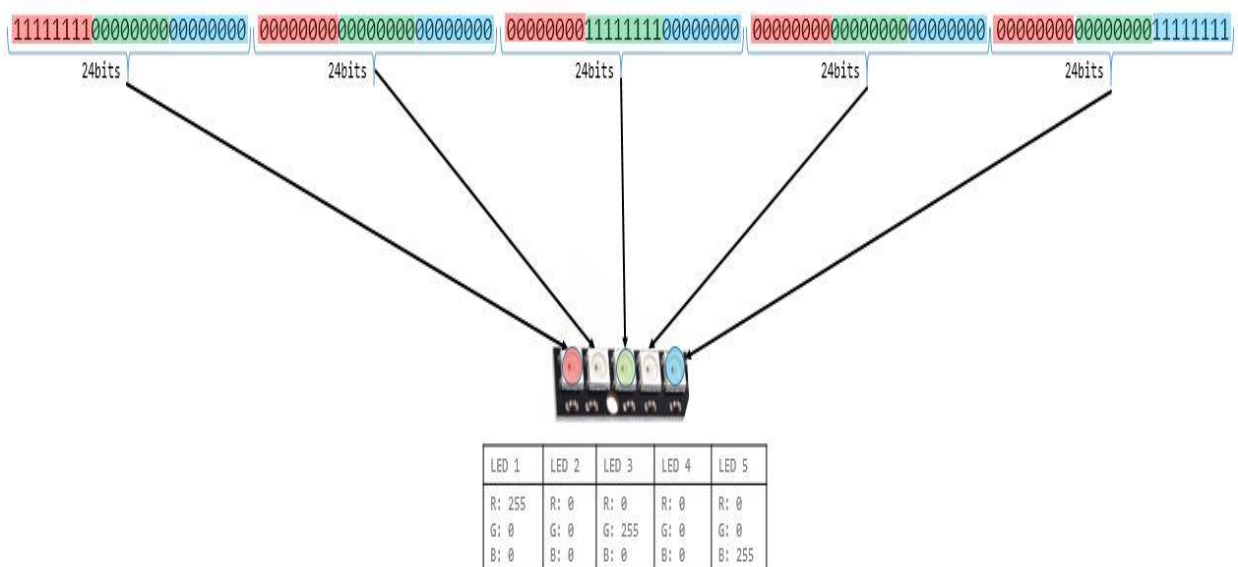


Figura 3. Funcionament de LEDs WS2812B

Això vol dir que sempre anem a enviar tota la informació per a cada un dels LEDs, encara que estiguin apagats.

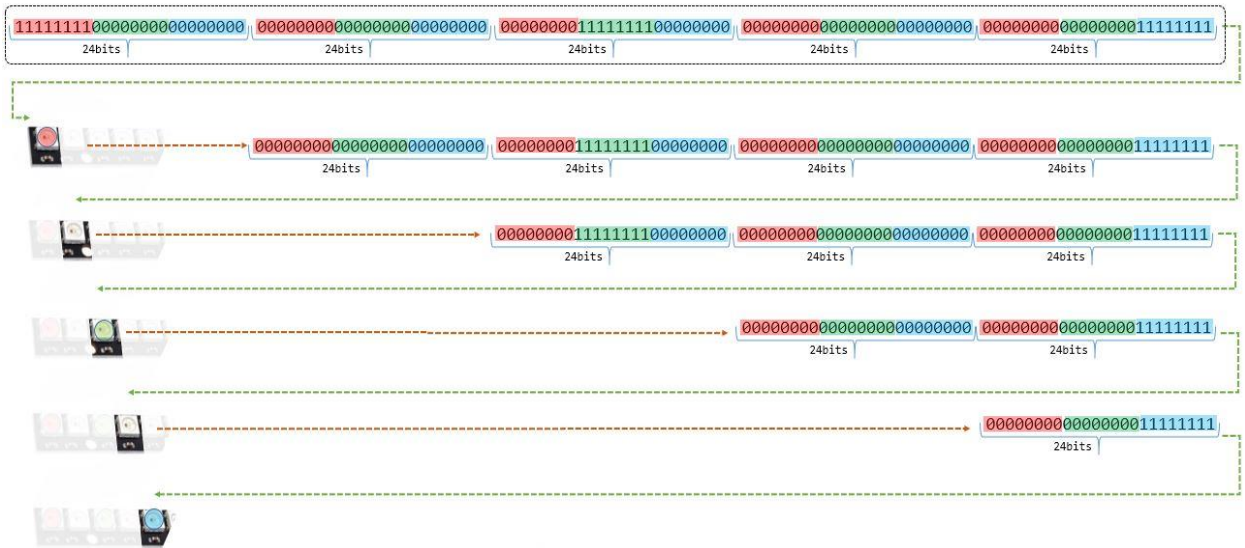


Figura 4. Enviades de dades de LEDs WS2812B

Després un cop tots tenen la informació, es llança el senyal resetcode i tots mostren la informació que tenen emmagatzemada:



Figura 5. Resultats

Aquesta genial idea permet fer configuracions de múltiples LED, en els quals únicament hem de comunicar-nos amb el primer d'ells i cada LED s'actua de transmissor de la seqüència als LED posteriors. A més permet que puguem encadenar o dividir tires de LED i qualsevol fragment seguirà funcionant perquè tots els LED tenen exactament el mateix comportament. A més cada LED quan transmet al següent senyal, realitza una reconstrucció, és a dir que d'aquesta forma no es distorsiona o s'acumula soroll per als següents LEDs. Això permet també poder alimentar tires de més de 5m sense necessitat de dispositius que amplifiquin el senyal.

2.2.3. Matriu 8X32 LED WS2812B flexible

Aquesta gran matriu de LED flexible de 32x8 píxels WS2812B (o "NeoPixel") és una bona manera d'afegir una quantitat impressionant de color mentre encara es pot controlar individualment cada LED. No estem fent broma, això és 256 LED direccionables individualment en un panell flexible de 320 mm x 80 mm. Es pot crear animacions, jocs o fins i tot incorporar-les en un divertit projecte de tèxtils electrònics. A més de tot això, gràcies a la seva flexibilitat de suport, aquesta LED matriu pot ser inclinada per adaptar-se a gairebé qualsevol superfície curvilínia.

Aquest panell matriu que mostra sota la figura 6, requereixen una entrada de baixa tensió de 5V per a la potència que necessita poder obtenir una bona quantitat de corrent, fins a 5A en la majoria dels usos (15A a pròximament, si tots els LEDs estan configurades en blanc brillant, el que no es recomana). Adjuntats a cada matriu LED flexible hi ha dos connectors JST-SM de 3 pins, un mascle i altre femella (entrada / sortida) que es proporcionen connexions per a DC5V+, DAT i GND. També té dos conjunts de cables d'alimentació, que es proporcionen connexions per a DC5V+, i GND.

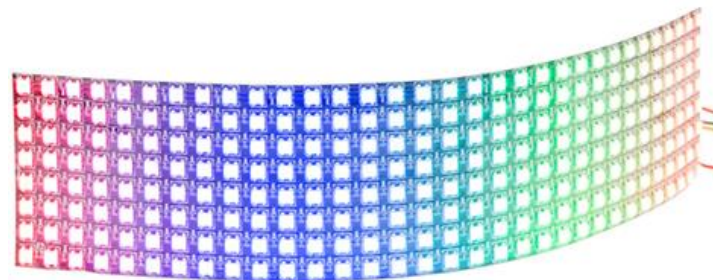


Figura 6. LEDs WS2812B matriu 8X32

Els matrius de LEDs no s'encenen sols, requereixen un microcontrolador (com Arduino) i una mica de programació. Llavors, a continuació, introdueix la programació pels matrius.

2.3. Programacions de les matrius LEDs

2.3.1. Introducció de NeoPixels

Els NeoPixels són una marca específica de LED RGB direccionables individualment venuts per Adafruit. "Adafruit Industries" és una companyia de maquinari de codi obert amb seu a la ciutat de Nova York.

Individualment direccionable significa que pot programar el quart LED a la tira per fer una cosa, mentre que la desena fa una altra cosa, i tots els altres fan una tercera cosa, etc. Hi ha moltes aplicacions divertides de Neopixels publicats a Internet amb diagrames de circuits i exemples de codi.

El protocol de control per NeoPixels es basa en un sol cable de comunicació. Adafruit proporciona una llibreria “Arduino” i una llibreria “Python” per ajudar amb la programació de NeoPixels que es veure al capítol 5.3.1 las instal·lacions al Python.

2.3.2. La llibreria de NeoPixels

La llibreria de la NeoPixel és per a control de píxels i tires de LED d'un sol cable, que es pot descarregar per l'entorn d'Arduino IDE.

La llibreria NeoPixel de Adafruit que és senzilla d'usar, contenen els exemples que venen incloses, perquè decideixin quins tipus d'efectes necessiten.

Hi ha diverses avantatges per utilitzar aquesta llibreria, i les més importants són, poden controlar els píxels d'una manera individual, també només necessiten un pin digital d'una placa microcontrolador per controlar-los a tots els LEDs. I poden controlar-los tant les intensitats com els color.

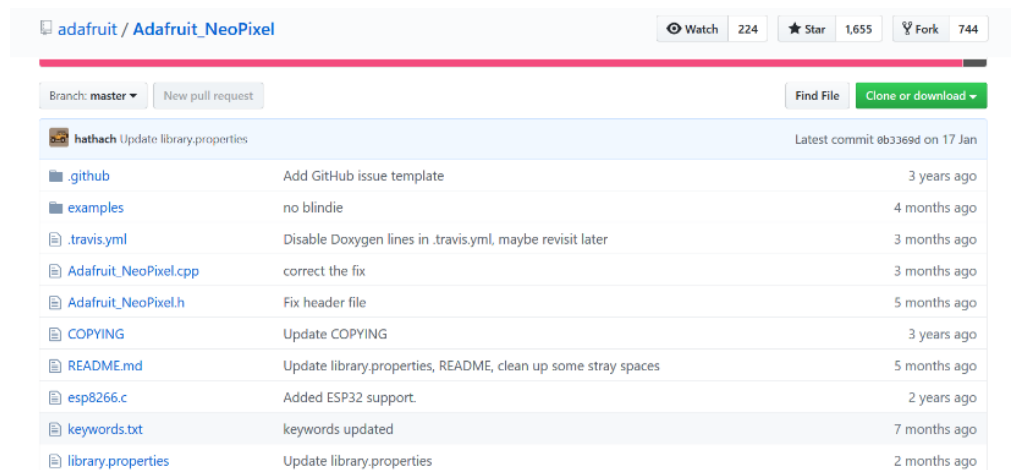


Figura 7. Adafruit_NeoPixel

Per utilitzar-la en qualsevol projecte sempre ha de reclamar-la en los primers encapçalats dels codis d'Arduino IDE, com les llibreries següents.

2.3.3. La llibreria de NeoMatrix

La llibreria “Adafruit_NeoMatrix” es basa en “Adafruit_NeoPixel” per crear pantalles gràfiques bidimensionals utilitzant “NeoPixels”. Després, pot dibuixar fàcilment formes, text i animació sense haver de calcular cada posició de píxel X / Y. Les pantalles més grans es poden formar utilitzant seccions de la tira “NeoPixel”, com es mostra a la Figura 8 de avall.

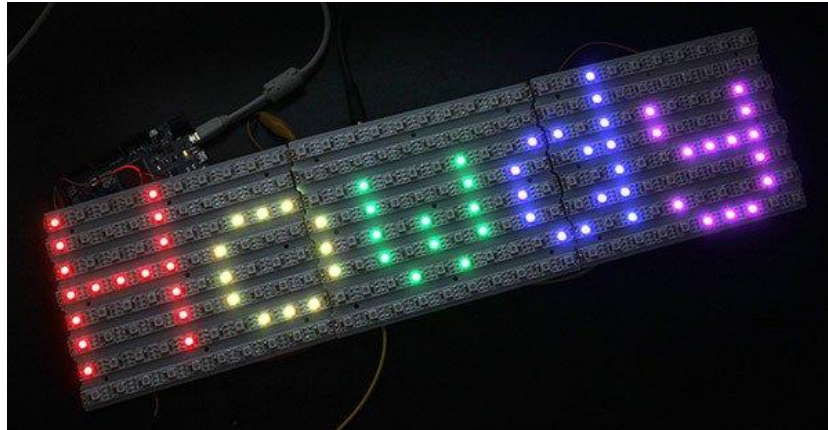


Figura 8. Adafruit_NeoMatrix

2.3.4. La llibreria Adafruit_GFX

La llibreria Adafruit_GFX de gràfics principal per a la majoria de pantalles, i proporciona un conjunt comú de primitius gràfics (punts, línies, cercles, etc.). Ha aparellar-se amb una llibreria específica de maquinari per a cada dispositiu de pantalla que es porta.

La llibreria Adafruit_GFX per Arduino proporciona una sintaxi comuna i un conjunt de funcions gràfiques per a totes les pantalles LCD i TFT. Això permet que els esbossos d'Arduino es puguin adaptar fàcilment entre els tipus de pantalla amb un mínim esforç, i qualsevol nova característica, millora de rendiment i correcció d'errors s'aplicaran immediatament en tota la oferta de pantalles en color.

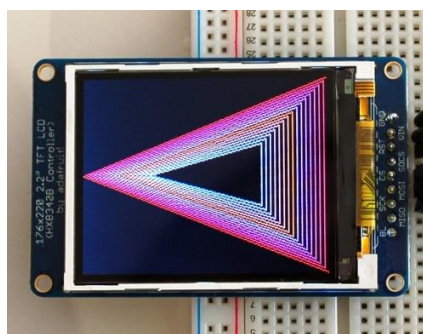


Figura 9. Adafruit_GFX

Al final, per funcionar bé les pantalles LEDs RGB com les lletres o les animacions de colors, els esbossos d'Arduino han d'incloure els tres encapçalats per utilitzar aquestes 3 llibreries sempre:

```
1. #include <Adafruit_GFX.h>
2. #include <Adafruit_NeoMatrix.h>
3. #include <Adafruit_NeoPixel.h>
```

Taula 1. Encapçalats 3 llibreres de Adaruit

2.4. Verificar i dissenyar el cablejat de fabricació de matrius

Els cablejats de les matrius de WS2812B de cada fabricants són diferents. Primer, ha de verificar en quina direcció o sentit van els LEDs quan fabriquen. Per exemple, On és el primer LED en un matriu i està en el part superior o inferior, i el següent píxel com va per la dreta o esquerra? És important, perquè ha de saber-los per escriure els codis de programació. O per les altres paraules aquí es centraren en el constructor: com declarar una pantalla bidimensional feta de "NeoPixels". La llibreria fa servir per les dues tipus de matrius, individuals que totes "NeoPixels" en una sola quadrícula uniforme, i la altre matrius en compleix, múltiples quadrícules combinades en una pantalla més gran.

"Adafruit_NeoMatrix" fa servir exactament el mateix sistema de coordenades, funcions de color i comandaments gràfics que la llibreria "Adafruit_GFX". Que es mostren per figura 10 de avall:

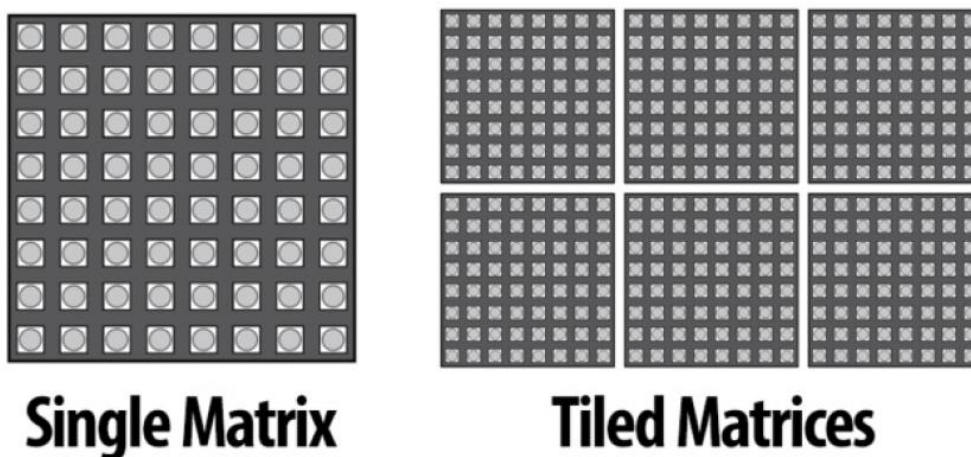


Figura 10. Matrius simple i compleix

2.4.1. Matriu simple

Comencen amb la declaració per a una sola matriu, perquè és més simple d'explicar. En aquest cas, es fa un exemple per demostrar una matriu de 5x8 de NeoPixels. Quan es mira aquest escut amb el text en una orientació llegible, el primer píxel, # 0, és a la part superior esquerra. Cada píxel endavant és correcte en una posició: el píxel #1 està directament a la dreta del píxel #0, i així successivament. Al final de cada fila, el següent píxel es troba a la banda esquerra de la següent fila. Això no és una cosa que van a decidir en el codi sinó estan cablejats a la placa de circuit que comprèn l'escut de fabricant el qual indiquen la figura 11 de sota.

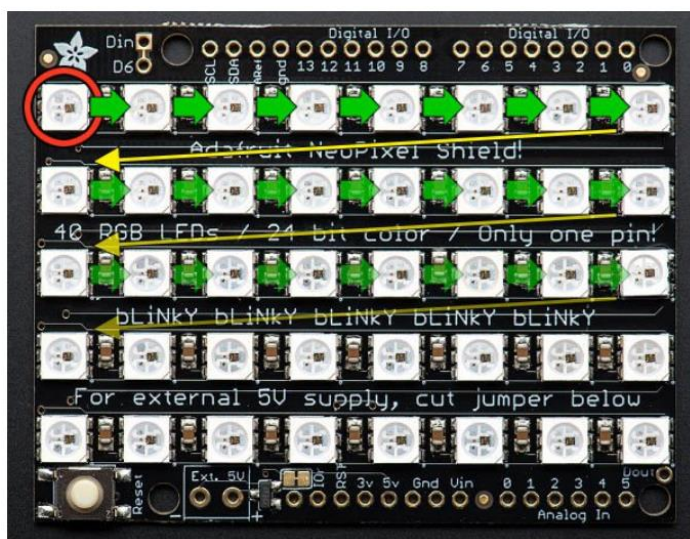


Figura 11. Matriu 5x8 de NeoPixels

Es referint a aquest disseny com fila major i progressiu. Fila major significa que els píxels estan organitzats en línies horitzontals (l'oposat, en línies verticals, és la columna més gran). Progressiu vol dir que cada fila avança en la mateixa direcció. Algunes matrius invertiran la direcció a cada fila, ja que pot ser més fàcil cablejar d'aquesta manera. D'això en diem un disseny en "zigzig".

Una vegada que saben l'orientació de la matriu, ja poden declarar els codis de la matriu amb Arduino IDE que es veu en la següent taula 2.

```

1 Adafruit_NeoMatrix matrix = Adafruit_NeoMatrix(5, 8, 6,
2   NEO_MATRIX_TOP + NEO_MATRIX_LEFT +
3   NEO_MATRIX_ROWS + NEO_MATRIX_PROGRESSIVE,
4   NEO_GRB + NEO_KHZ800);

```

Taula 2. Declaració matriu simple

Llavors, els primers dos paràmetres, “5” i “8”, són l'amplada i l'altura de la matriu, en píxels. El tercer paràmetre “6” és el nombre de pin al qual estan connectats els NeoPixels. En el blindatge, aquest està cablejat al pin digital “6”, però les matrius independents són lliures d'utilitzar altres pins. Al següent capítol ja es ho indicaran.

El següent argument és l'interessant. Això indica on es col·loca el primer píxel a la matriu i la disposició de files o columnes. El primer píxel ha d'estar en una de les quatre cantonades; en què cantonada s'indica afegint NEO_MATRIX_TOP o NEO_MATRIX_BOTTOM a NEO_MATRIX_LEFT o NEO_MATRIX_RIGHT. L'arranjament de fila / columna s'indica afegint addicionalment NEO_MATRIX_COLUMNS o NEO_MATRIX_ROWS a NEO_MATRIX_PROGRESSIVE o NEO_MATRIX_ZIGZAG. Tots aquests valors s'agreguen per formar un sol valor com en el codi anterior.

“NEO_MATRIX_TOP + NEO_MATRIX_LEFT + NEO_MATRIX_ROWS + NEO_MATRIX_PROGRESSIVE”. Això els codis que respecte al Figura 11.

L'últim argument és exactament el mateix que amb la llibreria “NeoPixel”, que indica el tipus de píxels LED que s'estan utilitzant. NEO_GRB vol dir que els píxels estan cablejats per al flux de bits verd-vermell-blau, la majoria dels productes segueix en aquest tipus. I un altre NEO_RGB s'utilitza per píxels que estan cablejats per vermell-verd-blau. NEO_KHZ800 és la freqüència del flux de bits, en cas de WS2812B s'utilitza per píxels amb un flux de bits de 800 kilohertz, i NEO_KHZ400 s'utilitza per píxels amb un flux de bits de 400 kilohertz. En la majoria dels casos amb els últims productes de “NeoPixel” o altres productes similars, simplement pot deixar aquest argument fos, i el codi d'exemple és simplement descriptiu.

2.4.2. Matriu complexa

Una matriu complexa és la combinació de diverses matrius petites. Això és de vegades més fàcil per al muntatge o per a distribuir l'energia. Totes les submatrius han de ser de la mateixa mida i s'han d'ordenar de manera previsible. La llibreria de “Adafruit_NeoMatrix ()” rep alguns arguments addicionals:

```
1. Adafruit_NeoMatrix matrix = Adafruit_NeoMatrix (  
2.   matrixWidth , matrixHeight , tilesX , tilesY , pin , matrixType , ledType );
```

Taula 3. Declaració matriu complexa

Els primers dos arguments són el longitud i l'amplada de cadascun de submatriu, no tota la pantalla.

Els següents dos arguments són el nombre de submatriu, en la direcció horitzontal i vertical. Les dimensions de la visualització general seran sempre un múltiple de les dimensions de la submatriu.

El cinquè argument és el nombre de pin, igual que abans i amb la biblioteca NeoPixel. L'últim argument també segueix comportaments anteriors i, en la majoria dels casos, es pot deixar de banda.

Amb una sola matriu, hi havia una cantonada inicial, un eix major (files o columnes) i una seqüència de línies (progressiva o en ziga-zaga). Ara s'ha duplicat, però es necessita informació similar. Tant per a l'ordre de píxels dins dels mosaics individuals, com per la disposició general dels mosaics a la pantalla. Com abans, afegim una llista de símbols per generar un sol argument que descriu el format de visualització.

Els arguments NEO_MATRIX funcionen de la mateixa manera que en el cas anterior d'una sola matriu, i ara es refereixen a les submatrius individuals dins de la pantalla general. Totes les fitxes han de seguir el mateix format. Un conjunt addicional de símbols funciona de manera similar per després descriure l'ordre dels mosaics.

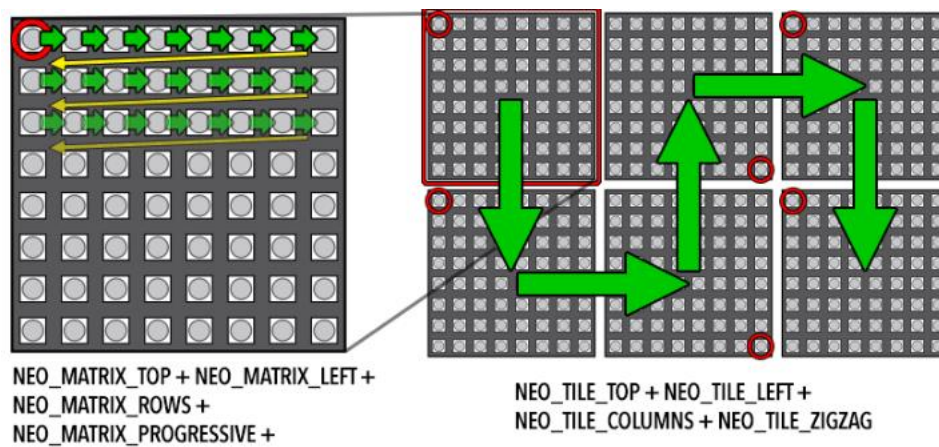
El primer mosaic ha d'estar ubicat en una de les quatre cantonades. Afegeix NEO_TILE_TOP o NEO_TILE_BOTTOM i NEO_TILE_LEFT o NEO_TILE_RIGHT per indicar la posició del primer rajola. Això és independent de la posició del primer píxel dins dels mosaics; Poden ser cantonades diferents.

Els mosaics es poden organitzar en files horitzontals o columnes verticals. Novament, això és independent de l'ordre de píxels dins dels mosaics. Afegeix NEO_TILE_ROWS o NEO_TILE_COLUMNS.

Finalment, les files o columnes de mosaics es poden organitzar en ordre progressiu o en ziga-zaga; és a dir, cada fila o columna procedeix en el mateix ordre, o alternant les files / columnes canviant de direcció. Afegeix NEO_TILE_PROGRESSIVE o NEO_TILE_ZIGZAG per indicar la comanda. Però, si es selecciona l'ordre NEO_TILE_ZIGZAG, les línies alternes de mosaics han de ser girat 180 graus. Això és intencional i per disseny; Manté el cablejat de rajola a rajola més consistent i simple. Aquesta rotació no és necessària per NEO_TILE_PROGRESSIVE.

2.4.3. Combinació total de matrius

Finalment, faran la combinació total de matrius amb seus codis que respecte al figura 12.



```

1 Adafruit_NeoMatrix matrix = Adafruit_NeoMatrix(8, 8, 3, 2, PIN,
2   NEO_TILE_TOP + NEO_TILE_LEFT + NEO_TILE_COLUMNS + NEO_TILE_ZIGZAG +
3   NEO_MATRIX_TOP + NEO_MATRIX_LEFT + NEO_MATRIX_ROWS + NEO_MATRIX_PROGRESSIVE,
4   NEO_GRB + NEO_KHZ800);

```

Figura 12. Combinació total de matrius

Un cop, es coneix la funcionalitat de matrius LEDs RGB amb la seva llibreria de Adafruit, al següent capítol es concentren les coneixement de microcontrolador per controlar les matrius LEDs RGB.

3. NODEMCU EL KIT DE DESENVOLUPAMENT

3.1. La selecció de MCU amb seu kit de desenvolupament

En el capítol anterior deia per funcionar bé les matrius LEDs RGB també necessita un control com microcontrolador, per enviar instruccions des d'aquest microcontrolador que indica al matriu LEDs quan encendre i quins colors encendre etc. Llavors, primer ha de fer una selecció sobre en el mercat de microcontrolador actual i les seves plaques de desenvolupament. Hi ha un gran nombre d'opcions de microcontrolador per operar matrius LEDs, però d'aquí es comparen 3 microcontroladors amb la seva placa desenvolupament que són molt habitual en el mercat actual.

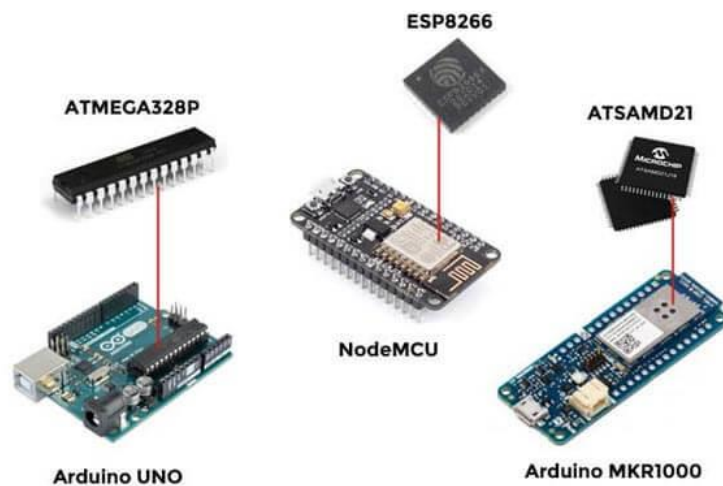


Figura 13. Tres plaques diferents de desenvolupaments

Aquí no es repeteixen les introduccions ni les característiques de microcontroladors bàsics. L'objectiu és programar la MCU o microcontrolador mitjançant el kit o placa de desenvolupament. Tota la resta se serveix de suport perquè crear propis projectes sigui el més senzill possible.

Les plaques com Arduino UNO o Arduino MKR1000 utilitzen la mateixa arquitectura això sí, cadascuna amb les seves característiques particulars.

La major avantatge que tenen plaques com Arduino MKR1000 i NodeMCU és que incorporen un mòdul WiFi que ens permet crear projectes del IOT o sistemes sense fil.

Són el primer pas cap a l'Internet de les Coses o el IOT. Poden enviar dades, rebre dades i fins i tot controlar els pins d'entrada i sortida de forma remota i sense fil.

La gran diferència entre el ESP8266 i el ATSAMW25 és el preu. Es veure la taula següent.

Plaques de desenvolupaments	Soc	Wifi	Preu(€)
NodeMCU	ESP8266	SÍ	8
Arduino MKR1000	ATSAMW25	SÍ	52
Arduino ONE	ATMEGA328P	NO	18

Taula 4. Comparatives de plaques de desenvolupament

Al final es selecciona el kit NodeMCU 1.0 per fer el projecte que és una placa de desenvolupament totalment oberta, a nivell de programari i de maquinari, també pel seu bon preu, i bones prestacions a usar en el projecte LEDs REG. Està muntada al voltant del conegut xip ESP8266 (el qual ofereix una solució completa i autònoma de xarxes Wi-Fi, el que li permet allotjar l'aplicació o servir com a pont entre Internet i un microcontrolador) i exposa tots els seus pins en els laterals. Al següent apartat comença a conèixer NodeMCU 1.0 de versió V3 des del base en un Soc ESP8266. D'altra banda, es ofereix més avantatges com la incorporació d'un regulador de tensió integrat, així com un port USB de programació. Es pot programar amb LUA o mitjançant l'IDE d'Arduino.

Un petit repàs de que cal no confondre microcontrolador amb placa de desenvolupament. NodeMCU 1.0 no és un microcontrolador de la mateixa manera que Arduino MKR1000 tampoc ho és. Són plaques o kits de desenvolupament que porten incorporats un xip que se sol anomenar SoC (System on a Chip) que dins té un microcontrolador o MCU. L'esquema general d'aquest tipus de plaques seria el següent.

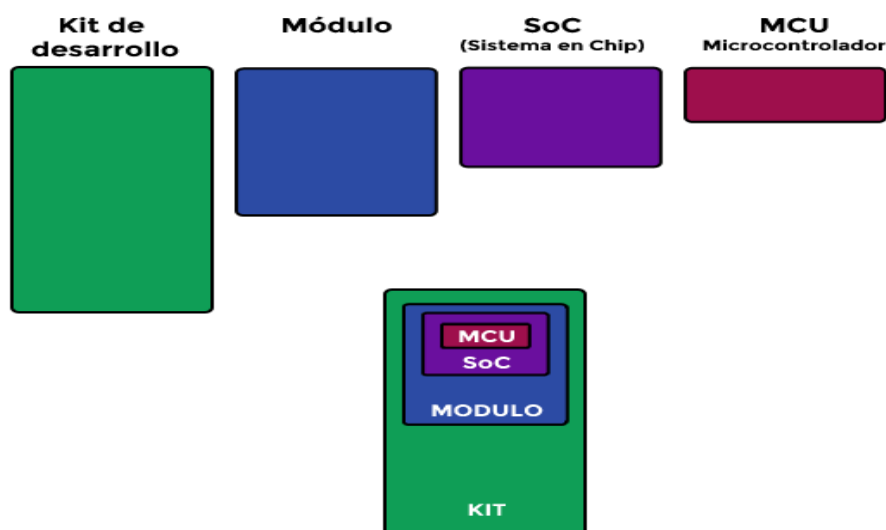


Figura 14. L'esquema general tipus de plaques

3.2. La introducció del SoC ESP8266

El ESP8266 és un xip que se sol anomenar SoC que dins té un microcontrolador o MCU. I aquest xip Wi-Fi de baix cost amb pila TCP / IP, amb capacitats de 2.4 GHz Wi-Fi (802.11 b / g / n, suport WPA / WPA2), completa i capacitat de MCU produïda pel fabricant xinès Espressif Systems, amb seu a Xangai.

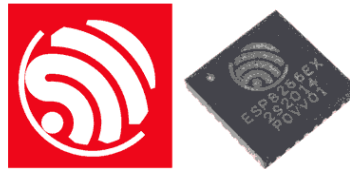


Figura 15. SoC ESP8266

El ESP8266 té potents capacitats a bord de processament i emmagatzematge que li permeten integrar-se amb sensors i dispositius específics d'aplicació a través dels seus GPIOs amb un desenvolupament mínim i càrrega mínima durant el temps d'execució. El seu alt grau d'integració en el xip permet una circuiteria externa mínima, i la totalitat de la solució, incloent el mòdul està dissenyat per ocupar l'àrea mínima en un PCB.

Per tant, en termes estrictes l'ESP8266 no és un microcontrolador. Dins sí que porta un CPU i es diu Tensilica L106 de 32 bits amb arquitectura RISC que funciona a una velocitat de 80MHz, amb una velocitat màxima de 160Mhz. La MCU es va a encarregar de gestionar totes les entrades, sortides i càlculs necessaris per fer funcionar el programa que hàgim carregat.

El nom tècnic és ESP8266EX, amb seu diagrama de bloc funcional demostra la següent figura.

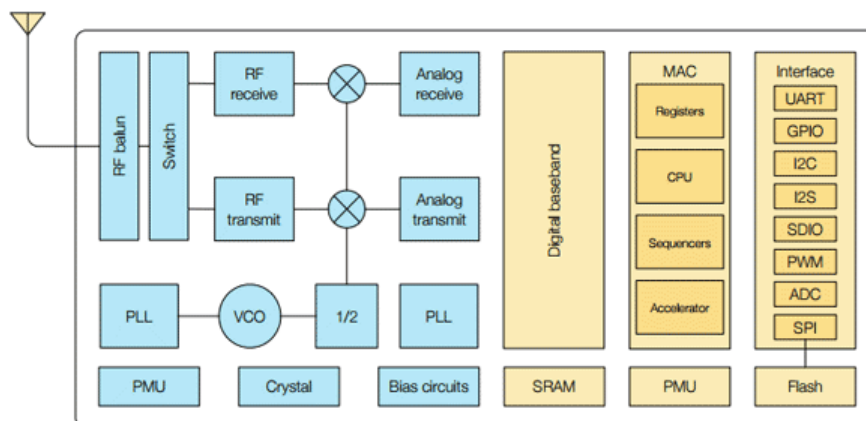


Figura 16. Diagrama de bloc funcional

Funciona amb 32 bits el que ve a dir que pot realitzar operacions amb nombres d'aquesta mida (de 0 a 4294967295 o de 2147483648 a 2147483647). No obstant això, les MCU més comuns són de 8 bits com la que porta el Arduino UNO l'ATmega328P.

Aquí té l'esquema de Pinut del ESP8266.

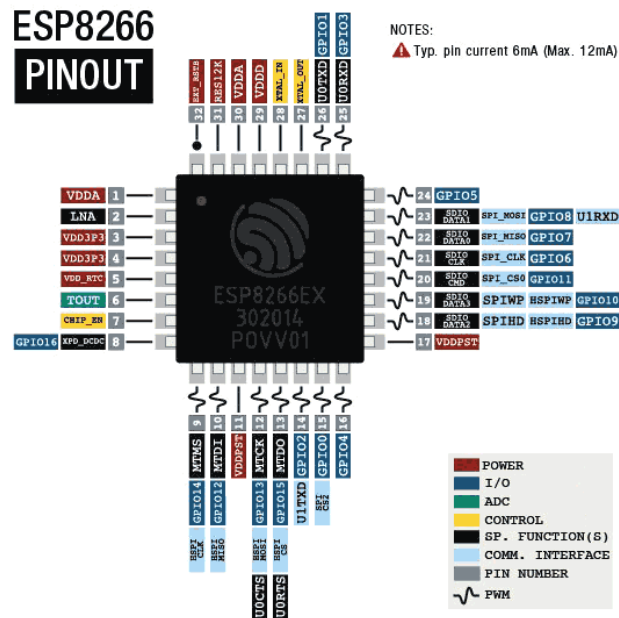


Figura 17. Pinout de ESP8266

El ESP8266 té 32 pins disponibles, dels quals 17 són GPIO, 1 és un ADC, i la resta essencialment estan relacionats amb l'alimentació i control del ESP8266.

Bàsicament consisteix en un xip que té tot integrat (o gairebé tot) perquè pugui funcionar de forma autònoma com si fos un ordinador. En el cas del ESP8266 l'únic que no té és una memòria per emmagatzemar els programes.

Això suposa un inconvenient ja que part dels pins d'entrada i sortida, hauran de ser utilitzats per a connectar-se a una memòria flash externa.

3.3. Mòduls ESP-12E

Els mòduls ESP8266 els poden trobar en diferents encapsulats i plaques, hi ha una àmplia gamma de mòduls basats en el SoC ESP8266. La diferència entre ells és l'accés als pins. Dependrà de cada mòdul tindràs accés a uns o altres pins que es veure la figura 18.



Figura 18. Gamma de mòduls de ESP8266

Dins dels diferents mòduls del ESP8266, es troba ESP-12E. Aquest mòdul és un dels més usats, té disponible tots els pins que el xip ESP8266 posseeix. És a dir, al ESP12E tenen disponibles la majoria de pins importants del SoC, mentre que el mòdul s'encarrega de connectar elèctricament la majoria de pins d'alimentació i control. Aquest és un dels motius de l'èxit i alta implementació del ESP12E, ja que s'ofereix totes les funcions del ESP8266. Bàsicament aquest mòdul incorpora 4 MB de la memòria flash per emmagatzemar els programes o sketches i l'antena.

D'altra banda, el ESP12E és el mòdul amb SoC ESP8266 més emprat en plaques de desenvolupament. Entendre el seu Pinout està molt relacionat amb entendre el ESP8266, prescindint de pins que no s'aporten massa (interns).

Aquí té l'esquema de Pinout del ESP12E.

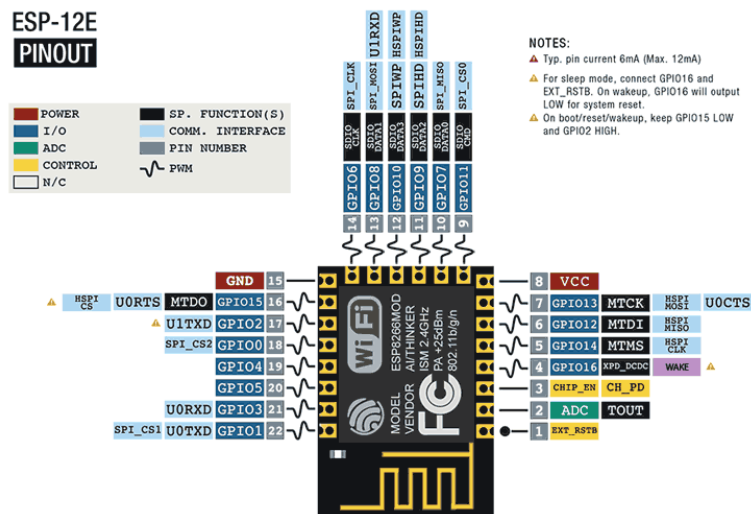


Figura 19. ESP-12E Pinout

Com es veuen la figura dalt, el ESP12E té 22 pins. Porten els 17 pins GPIO, el pin ADC, els pins d'alimentació a 2 (Vcc i GND) i els de control a CHIP_EN (encendre i apagar el ESP12E) i EXT_RSTB (per fer Reset).

Però encara és complicat per poder programar aquest tipus de mòduls, pel seu format és necessari un adaptador, crear un PCB o incorporar un kit desenvolupament. Així introdueix el kit desenvolupament de NodeMCU V1.0 que base en el mòdul ESP-12E que es veuen al següent apartat.

3.4. Kit o placa de desenvolupament NodeMCU

Ara arriba al nivell superior on tot és molt més fàcil. Aquest tipus de kit incorpora components que es ajuden a programar i connectar el mòdul i la MCU als circuits que serà molt més senzill. Hi ha diferents models de diverses marques amb característiques i funcionalitats diferents depenent del SoC o microcontrolador que utilitzen. Però tots tenen el mateix objectiu, es facilitaran el prototipat i desenvolupament de projectes amb microcontroladors.

En aquest nivell, ja no es preocupen de com carregar el programa o com connectar els pins. Ara només necessitem un ordinador, un cable micro-USB i un entorn de desenvolupament com Arduino IDE per a programar la MCU o microcontrolador.

3.4.1. Versions de NodeMCU

Hi ha una gran confusió pel que fa a les diferents versions que hi ha de NodeMCU. Tot això és degut a que es tracta d'una placa de maquinari obert i qualsevol fabricant pot crear la seva pròpia distribució.

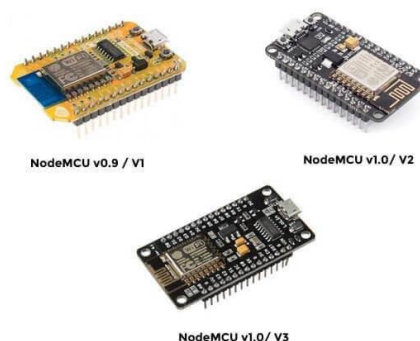


Figura 20. Versions de NodeMCU

Actualment hi ha 3 diverses versions en el mercat que es veu a la figura 20 de dalt. Existeix certa confusió al respecte i és a causa que s'han nomenat de diferents maneres. Poden trobar el nom depenent de la generació a la qual pertanyen, la versió o el nom comú que se'ls ha donat. El projecte es utilitza el kit de versió NodeMCU V1.0/V3 que diran a "NodeMCU V3".

La placa V3 és bàsicament la V2 amb algunes millores i pertany a la 2^a generació de NodeMCU V1.0 encara que no és una especificació oficial. Però, fer servir en Arduino IED només busqui la placa "NodeMCU V1.0 (ESP-12E Mòdul)" perquè son mateix generació 2^a, ja expliquin a la següent subapartat sobre el NodeMCU V3.

3.4.2. Accés als pins de NodeMCU V3

Un cop han vist què és el NodeMCU i les versions que existeixen, es comença descrivint les característiques més importants que s'aporta aquesta placa a nivell de maquinari, els pins digitals, analògics i d'alimentació.

A la següent figura es pot veure una visió general de tots els pins de NodeMCU V1.0/V3, que es muntarà en la pantalla LEDs RGB.

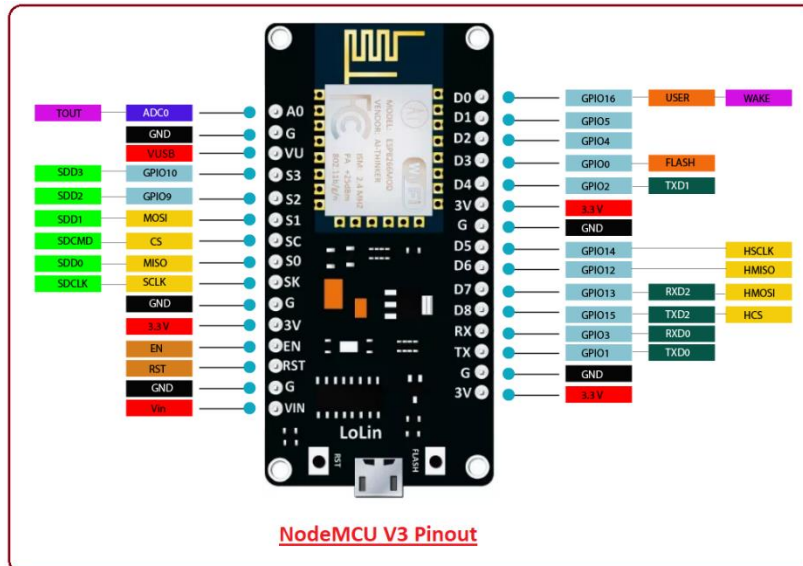


Figura 21. Pins de NodeMCU V3

Com a norma, sempre que faci referència a aquest diagrama de pins de NodeMCU, equivaldrà a si es posen la placa amb el port USB. Gràcies a la placa de desenvolupament NodeMCU poden accedir als pins d'una forma senzilla. El més pràctic és punxar la placa en una protoboard i des d'aquí, treure cables per connectar-los als diferents sensors i components.

3.4.3. Pins digitals de NodeMCU V3

Els pins digitals de NodeMCU V3 van numerats de l'D0 al D8, total són 9 pins digitals per utilitzar. Pins digitals de NodeMCU V3 es veu a la següent figura 22.

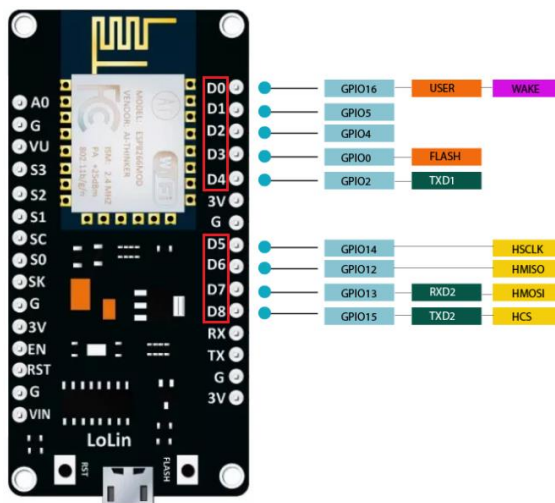


Figura 22. Pins digitals

Es troben 2 nomenclatures per nomenar els pins, els que apareixen a la placa escrits i en ocasions es veu el nom associat a cada Dx amb GPIOx. D'altra paraula, el nom dels pins de NodeMCU té una nomenclatura diferent de la que utilitza el ESP8266. Per exemple els pins D0, D1, i D2 en la NodeMCU V3 corresponen amb GPIO16, 5 i 4 al ESP8266. Per l'Arduino IDE permet utilitzar totes dues per referir-se als pins d'entrada/sortida, dues sentències són equivalents que es veuen a la següent figura.

```

Acceso a los pines con la nomenclatura de NodeMCU
1 pinMode(D2, OUTPUT);
2 digitalWrite(D2,HIGH);

Acceso a los pines con la nomenclatura de ESP8266
1 pinMode(4, OUTPUT);
2 digitalWrite(4,HIGH);
    
```

Figura 23. Dues nomenclatures equivalents

3.4.4. Pin analògic de NodeMCU V3

Si abans parlaven que en els pins digitals només hi han dos estats LOW i HIGH, al pin analògic va poder tenir un rang de valors. Aquest rang vindrà determinat per la resolució del convertidor ADC (convertidor analògic digital).

El NodeMCU V3 té només un pin analògic que admet un rang de valors de 0 a 3,3V amb una resolució de 10 bits.

Això implica que dins el codi tindrà un valor entre 0 i 1023 que es mapeja amb el voltatge entre 0 i 3,3V que s'indica a la figura 24.

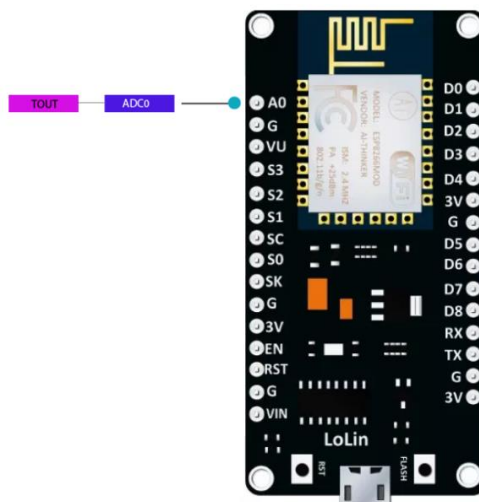


Figura 24. Pin analògic

3.4.5. Pins d'alimentació de NodeMCU V3

Els pins d'alimentació tenen 2 funcions: alimentar sensors i components (sortida), altra alimentar la pròpia placa (entrada).

El voltatge d'operació de NodeMCU V3 és de 3,3V i per tant, en principi no podrien alimentar cap component que necessités 5V.

No obstant això, quan s'alimenta a través del port USB amb 5V, internament té un regulador de voltatge que treu 3,3V i 5V. Els 3,3V s'utilitzen per alimentar el NodeMCU (electrònica). Els 5V s'utilitzen per alimentar altres components dins de la placa i per treure'ls pel pin de 5V. A més, per qualsevol d'aquests pins podem subministrar el mateix voltatge permetent així alimentar la placa a més del port USB. Això sí, si alimentem amb 3,3V per algun dels pins marcats amb aquest valor, la sortida de 5V ja no es subministrarà aquests 5V. Això cal tenir-ho en compte si en el projecte hi ha sensors que s'alimenten amb 5V.

En cas de Vin, és un regulador de voltatge que ve amb la capacitat de suportar fins a 800 mA. Pot manejar en algun lloc entre 7 i 12 V. No pot alimentar els dispositius que funcionen a 3.3V, ja que aquest regulador no pot generar tan sols 3.3 V.

Hi ha una diferència franca entre Vin i VU, on primer és el voltatge regulat que pot estar en algun lloc entre 7 i 12 V, mentre que més endavant és el voltatge d'alimentació per USB que s'ha de mantenir al voltant de 5 V.

Llavors, té 3 pins de 3,3V, 2 pin de 5V, 5 pins de terra GND en la versió V3 que s'indica a la figura 25.

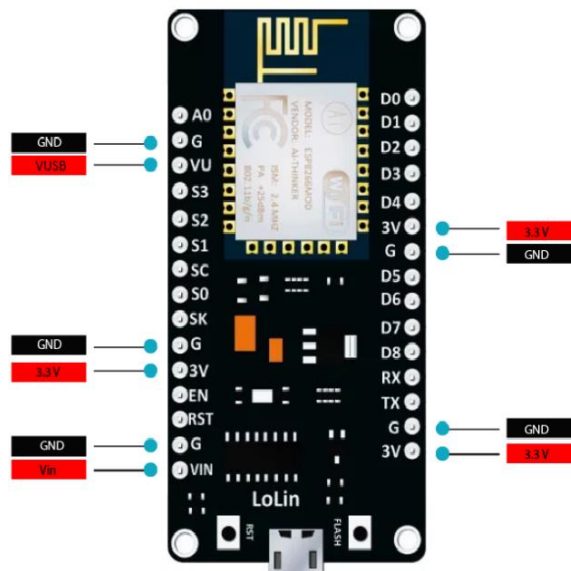


Figura 25. Pins d'alimentació

3.4.6. LEDs i pulsadors de NodeMCU V3

Ja se coneixen els pins d'entrada i sortida digitals, el pin analògic i els pins d'alimentació de NodeMCU. A més de tot això, aquesta placa es ofereix uns components extra per controlar el seu funcionament.

Dins de la pròpia placa de desenvolupament trobarem 2 LEDs integrats i 2 pulsadors.

El NodeMCU V3 té dos LEDs integrats. Cadascun d'ells estan associats als pins D0 (GPIO16) i D4 (GPIO2). Això vol dir que des del codi podran encendre i apagar cada un d'ells.

El LED connectat al D4, pertany al mòdul ESP-12E i el LED connectat al D0 és de la pròpia placa. Per encendre'ls han de posar un estat LOW (0V) i per apagar els LEDs han de posar un estat HIGH (3,3V).

A més de poder-los encendre i apagar a través del codi, el LED connectat al D4 ens serveix per saber si s'està carregant un programa. Cada vegada que es donen al botó de pujar, mentre dura la transmissió del programa el LED parpelleja.

El típic botó de RESET (RST) que si ho premen que es restableix la placa i comença l'execució de zero i un altre botó que posa FLASH.

El botó de RESET fa això, resetejar. Això no vol dir que elimini el codi, l'únic que fa és començar l'execució des del principi passant per la funció "setup ()" en entorn de l'Arduino.

El botó de FLASH es permet carregar un programa o microprogramari. Això no és una cosa específic de NodeMCU o del ESP8266, tots els microcontroladors tenen com a mínim dos estats.

L'estat d'execució és quan el microcontrolador executa el programa que han carregat. L'estat càrrega de programa o de microprogramari ens permet pujar un programa al microcontrolador. Tots aquests LEDs d'estat i els polsadors es veuen a la figura 26.



Figura 26. LEDs i polsadors de placa indicats

Fins ara es coneixen el kit desenvolupament de NodeMCU V3 per controlar les matrius LEDs RGB direccionables per teòricament. I des de capítol següent faran totes les preparacions, com els programari o els suports per funcionar el sistema de pantalla dinàmica gestionada de manera remota.

4. PREPARACIÓ DEL ARDUINO IDE PER ESP8266

4.1. La instal·lació del controlador

Primer de tot, ha d'instal·lar el controlador. Connecta el NodeMCU 1.0 / V3 al ordinador a través d'un cable micro-USB. Igual que amb la majoria de les altres plaques, driver CH340-Xip s'utilitza per a la comunicació, que Windows reconeix automàticament.

Després de tornar a connectar-lo, aquest s'ha de reconèixer com un dispositiu "USB-SERIAL CH340" (Windows).

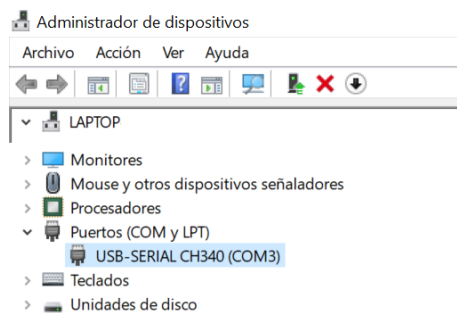


Figura 27. Instal·lació del controlador

4.2. La instal·lació d'Arduino IDE en sistemes Windows

Per començar, es descarrega el IDE d'Arduino, de la pàgina oficial d'Arduino, en el següent enllaç. Visiteu la pàgina "<https://www.arduino.cc/en/Main/Software>" i descarregui l'última versió per al seu sistema operatiu. Alternativament, pot registrar-se en Arduino Web-Editor i seguir les instruccions fàcils d'entendre proveïdes allà. Els següents primers passos fan servir la versió d'escriptori per a Windows. I executa el arxíu descarregat, llegeix i accepta la llicència de la instal·lació.

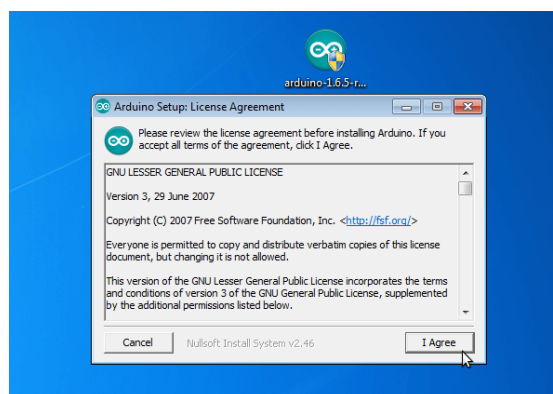


Figura 28. Descarrega i executa programa

La primera pantalla es veure la figura 29 que es sortirà es diràn que components vol instal·lar al costat del IDE, els deixem tots seleccionats i "Next" per continuar.

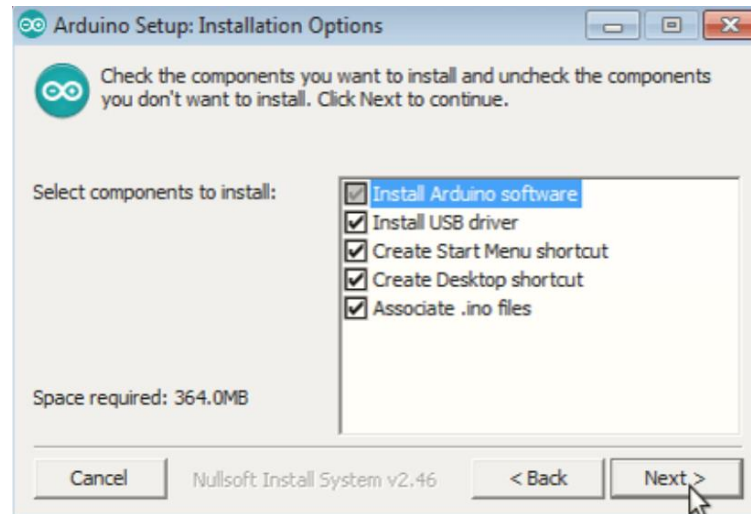


Figura 29. Selecció components

Ara es preguntarà en què ruta volen instal·lar l'IDE, poden triar la que vulguin i després de instal·lar per començar la instal·lació que es veure la figura 30.

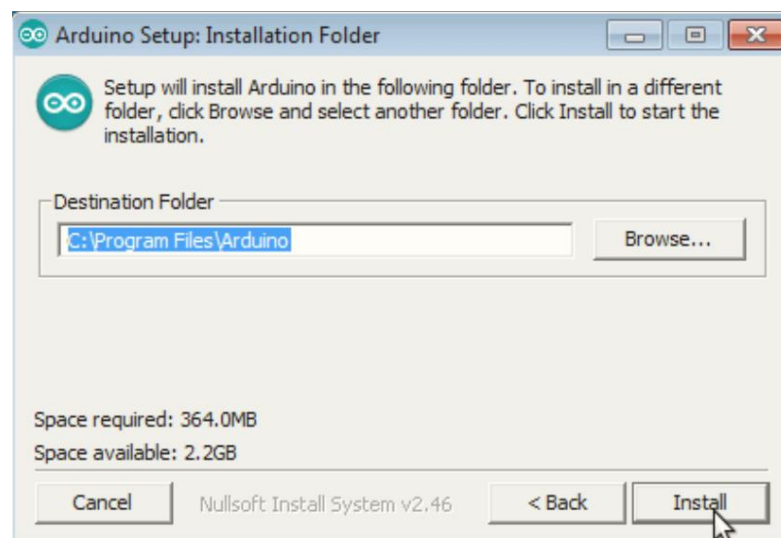


Figura 30. Instal·lació de Arduino IDE

Per finalitzar la instal·lació, Windows es preguntarà si vol instal·lar diversos Drivers (hi ha 4) per al funcionament dels microcontroladors per l'Arduino, recomani instal·lar tots, llavors, es veure en figura següent.

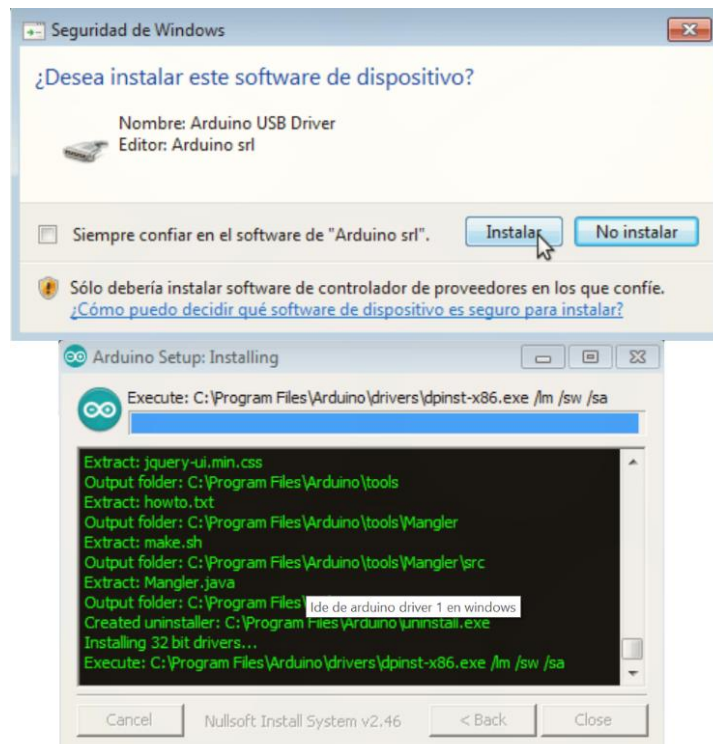


Figura 31. Instal·lacions de Drivers

Així, el programa es executa l'Arduino i es veure com s'inicia el IDE, a gaudir programant.



Figura 32. Iniciació de Arduino IDE

4.3. Afegir suport al IDE per ESP8266

El NodeMCU no és part del repertori estàndard del IDE, per aquesta raó, el board manager ha expandir-se. Llavors, de primer, ha d'obrir l'IDE d'Arduino, van entrar al menú arxiu.

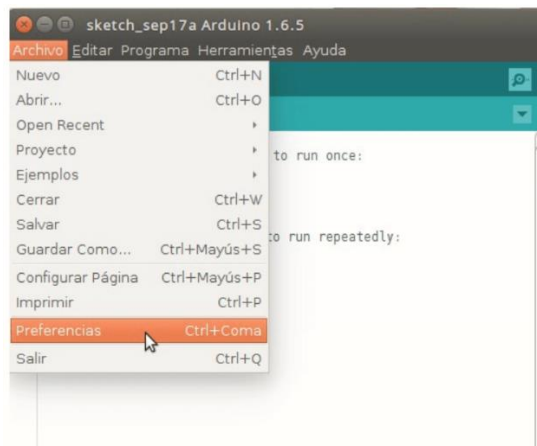


Figura 33. Anar al menú arxiu

Ara obre las preferències, i s'introdueix la següent url:

http://arduino.esp8266.com/stable/package_esp8266com_index.json, sense les cometes, a “additional boards manager url” i “ok” per tancar.

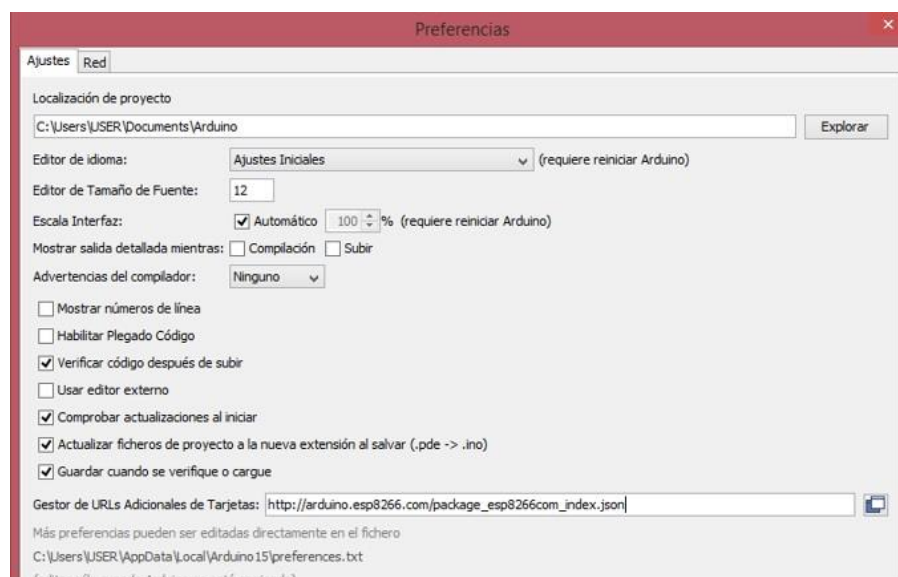


Figura 34. Copia de URLs de ESP8266

Després obre el menú eines, i dins l'opció placa, obre “board manager”. Dins de “board manager”, es veu una llista de suport per a plaques que pot instal·lar, si van fins al final veuen que surt l'opció esp8266, prem amb el ratolí sobre ella i es donarà l'opció per instal·lar. I és imprescindible que instal·lin la programació “esp8266 by ESP8266 Community”, es el suport de comunicació al ESP8266 que es veure a la figura següent.

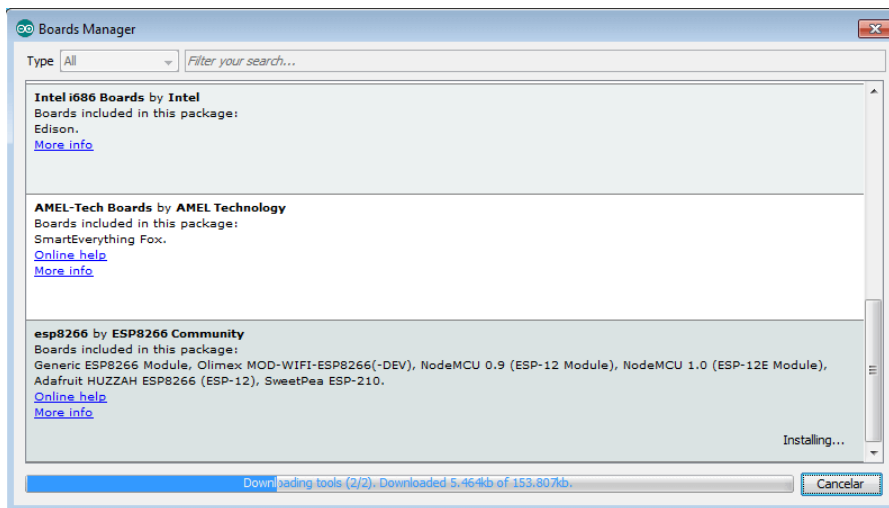


Figura 35. Instal·lació de suport

Finalment, un cop instal·lat el suport per ESP8266, ja només falta triar el model que vol programar, per això obre el menú eines, i dins l'opció placa al final del menú tindrà els mòduls ESP per triar. Llavors, selecciona la placa "NodeMCU 1.0 (ESP-12E Mòdul)" que havia explicat en el capítol anterior. Es veure a la següent figura 36.

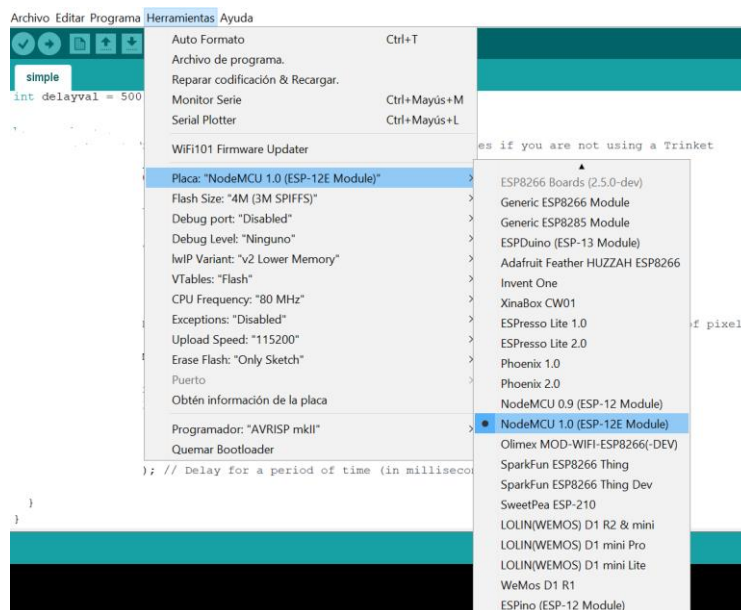


Figura 36. Selecció de ESP-12E

Un cop de configuren tots i es poden fer les proves amb programa de l'ESP-12E en l'Arduino IED per controlar els LEDs RGB. A la següent capítol introduirà com controlar els LEDs sense fil a través d'ESP8266.

5. PROGRAMANT EL ESP8266 PER VIA SENSE FIL.

El ESP8266 disposa d'un truc d'allò més interessant i important que consisteix a acceptar la programació OTA (Over The Air) o sigui, de manera sense fil, el que per descomptat obre moltes possibilitats per a l'IOT, perquè no sempre és fàcil poder anar a certs llocs amb el portàtil i un cable per reprogramarlos (i en moltes ocasions sol ser un fàstic). Per això la possibilitat de reprogramar un mòdul d'ESP8266 a través de WIFI, és una cosa molt d'agrar i dedicaran aquesta sessió a veure com fer-ho.

5.1. Instruccions de l'OTA

L'actualització de l'OTA (per sobre de l'aire) és el procés de carregar el firmware al mòdul ESP8266 mitjançant la connexió Wi-Fi en comptes d'un port sèrie. Aquesta funcionalitat es va fer extremadament útil en cas d'accés físic limitat o no al mòdul. Una de les característiques més interessants que tenen els ESP8266 és la capacitat de carregar remotament un firmware concret.

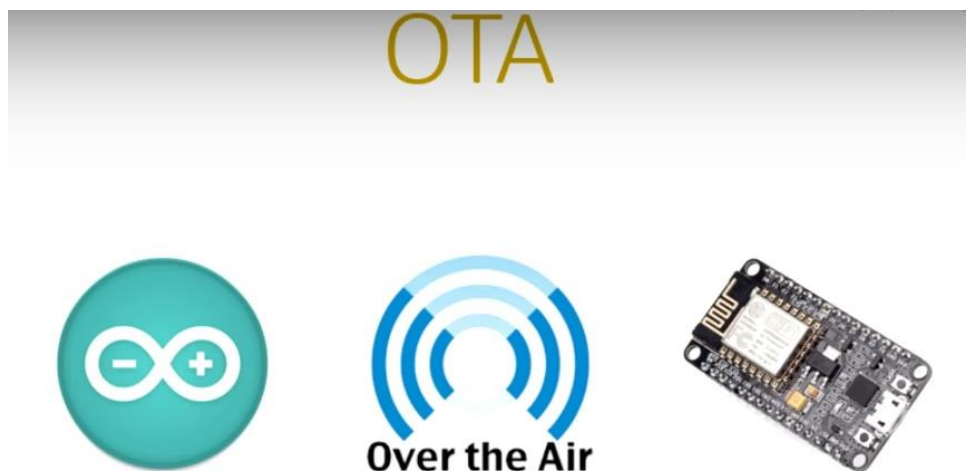


Figura 37. OTA

L'OTA es pot fer usant directe amb Arduino IDE, i també pot fer amb navegador web, o amb servidor HTTP.

En qualsevol cas, la primera càrrega del firmware s'ha de fer sempre a través d'un port sèrie. Si les rutines OTA s'implementen correctament en un esbós, totes les càrregues posteriors es poden fer a través de l'aire.

5.2. Seguretat bàsica de l'OTA

Per defecte, no hi ha cap seguretat imposada en el procés OTA. Depèn del desenvolupador per garantir que les actualitzacions només es permetin a partir de fonts legítimes o de confiança. Un cop finalitzada l'actualització, es reinicia el mòdul i s'executa el nou codi. El desenvolupador ha de garantir que la aplicació que s'executi al mòdul es tanca i es reinicia de forma segura. A continuació, proporcionen informació addicional sobre seguretat i seguretat del procés OTA.

El mòdul ha d'estar exposat sense fils per a actualitzar-se amb un nou esbós. Això planteja possibilitats que el mòdul sigui hacker violentament i carregat amb algun altre codi. Per reduir la possibilitat de ser hacker, consideri protegir els seus pujades amb una contrasenya, seleccionant cert port OTA, etc.

Certa funcionalitat de protecció ja està incorporada per la llibreria l'ArduinoOTA i no requereix cap codificació addicional per part del desenvolupador. Comprova la funcionalitat proveïda amb la llibreria ArduinoOTA que pot millorar la seguretat que es veure en següent taula 5.

```
1 void setPort(uint16_t port);
2 void setHostname(const char* hostname);
3 void setPassword(const char* password);
```

Taula 5. La funcionalitat de seguretat de ArduinoOTA

5.3. Requisits bàsics de l'OTA

La mida del xip flash ha de poder contenir el croquis antic (actualment en execució) i el croquis nou (OTA) al mateix temps.

S'ha de tenir en compte que el sistema d'arxius i la EEPROM, per exemple, també necessiten espai, en el NodeMCU 1.0 hi ha una memòria flash de 4M byte són suficients espais per a l'operació OTA.

La càrrega sense fil de mòduls des Arduino IDE està dissenyada per als següents escenaris típics: durant el desenvolupament del firmware com una alternativa més ràpida a la càrrega en sèrie, i per actualitzar una petita quantitat de mòduls, només si els mòduls ESPxx estan disponibles en la mateixa xarxa que l'ordinador amb Arduino IDE. Les instruccions a continuació mostren la configuració d'OTA a la placa NodeMCU 1.0 (mòdul ESP-12E).

5.3.1. Las instal·lacions

La instal·lació per OTA en base d'Arduino IDE amb entorn de ESP8266 també necessita una altra software per comunicar sense fil entre IDE a ESP8266 Board es diuen Python. Atenció, només pot instal·lar la versió 2.7.xx, no serveix les versions de 3.xx.

Els usuaris de Windows haurien de seleccionar "Afegir python.exe al camí" (vegeu més endavant: aquesta opció no està seleccionada per defecte).

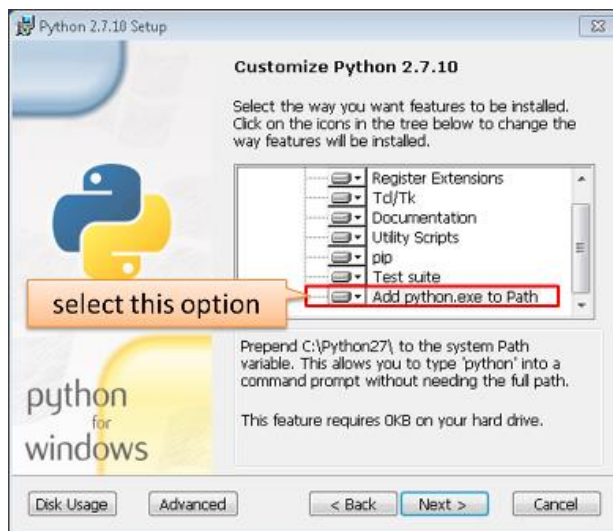


Figura 38. Selecció l'opció

5.3.2. Càrrega a través d'un port sèrie

Primer, inicia el Arduino IDE i descarreguen el programa des dels arxius. Actualitzin el SSID i la contrasenya en l'esbós perquè el mòdul pugui unir-se a la xarxa Wi-Fi. Es pot veure la següent figura.

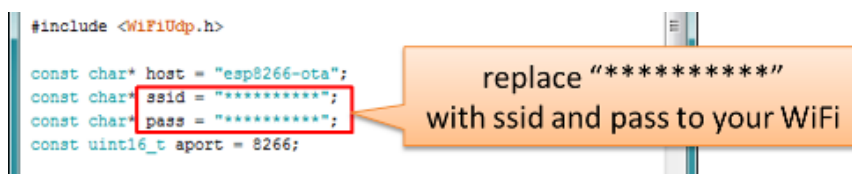


Figura 39. SSID i contrasenya de xarxa

Configura els paràmetres de càrrega com es mostra a continuació (és possible que s'hagi d'ajustar la configuració si esta utilitzant un mòdul diferent, ha de modificar-lo bé), com en la capítol anterior, però ara amb més detalls.

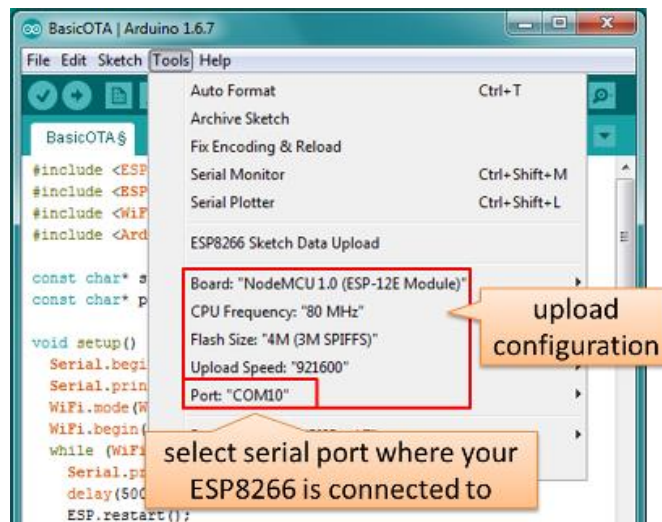


Figura 40. Configuració de NodeMCU 1.0

Carrega el croquis (Ctrl + U). Un cop fet això, obriu Serial Monitor (Ctrl + Maj + M) i verifiqui si el mòdul s'ha unit a la seva xarxa Wi-Fi. I amb comanda "cmd" es pot comprovar la funcionalitat de la xarxa de Wi-Fi. Tot això es comprova amb la figura 41.

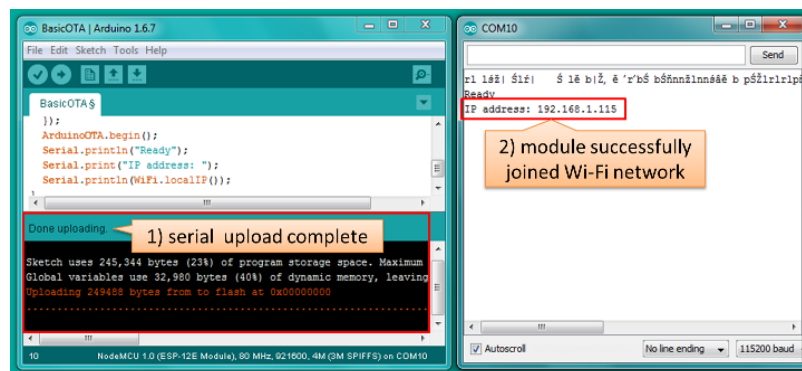


Figura 41. Càrrega i verifica de IP

El mòdul ESP s'hauria de restablir després de la càrrega en sèrie. En cas contrari, els passos posteriors no funcionaran. Es pot restablir automàticament després d'obrir el monitor sèrie com a visible a la captura de pantalla anterior.

5.3.3. Troba el port OTA

Només si el mòdul està connectat a la xarxa, després d'un parell de segons, el port esp8266-ota apareixerà en Arduino IDE. Selecciona el port amb l'adreça IP que es mostra a la finestra del Monitor sèrie en el pas anterior.

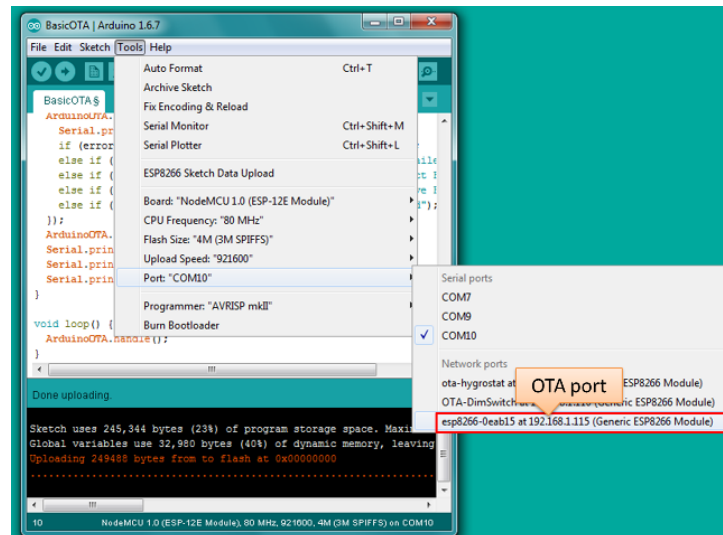


Figura 42. Troba el port d'OTA

Si el port OTA no es mostra, surti de l'IDE d'Arduino, obri'l de nou i verifiqui si el port està allà. Si no funciona, comproveu la configuració del seu Firewall i router.

Ara preparés per a la seva primera càrrega OTA en el qual que seleccionant el port OTA.

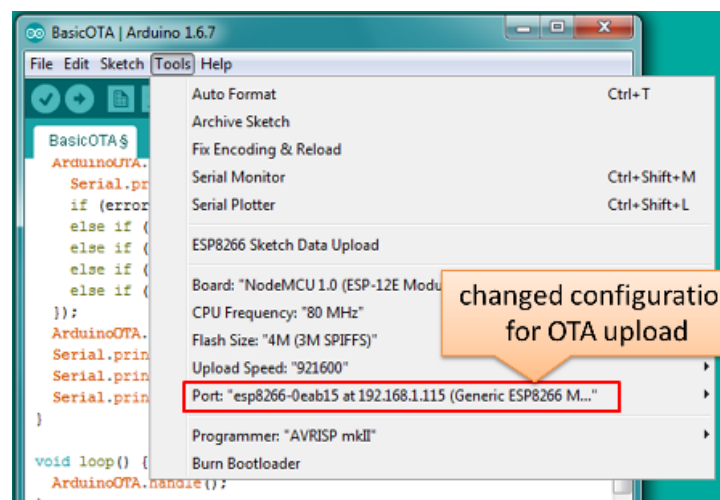


Figura 43. Selecciona la port OTA

5.3.4. Recàrrega nova amb OTA

A continuació, intenta carregar l'esbós de nou (utilitzant OTA). Una vegada que es completa la compilació, una vegada que la càrrega està a punt de començar, hauria de veure la sol·licitud de contrasenya de la següent manera.

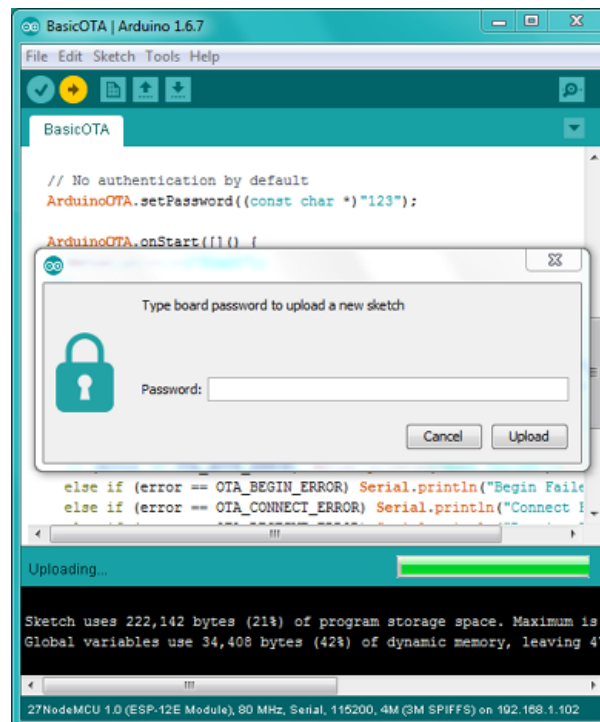


Figura 44. La sol·licitud de contrasenya

Introduïu la contrasenya i la càrrega s'ha d'iniciar la missatge visible en el registre de càrrega que es veure a la figura 45.



Figura 45. Autenticació i càrrega completament

Un cop connecta el port OTA i càrrega completament, ja puguin pujar el nou esbós des Arduino IDE amb la modificació de codis, o qualsevol altre esbós sobre OTA.

6. FONTS D'ALIMENTACIÓ PER LA PANTALLA LEDS RGB

Des del capítol 2 fins ara s'explicava la teoria d'una pantalla LEDs RGB ha estat controlat per un kit desenvolupament NodeMCU V3 a través d'un cable micro-USB i al final gestiona de manera remota per OTA.

En canvi, per funcionar bé tots els circuits com la pantalla o el NodeMCU, també necessita una bona font d'alimentació per garantir el sistema que funcionin contínuament.

6.1. Fonts d'alimentació

Així, primer s'introdueix el càlcul de la potència de la pantalla. Segons el datasheet cada mòdul de matriu de 8x32 LEDs RGB porten total 256 elements i cada LED com un píxel que, en pitjor cas, vol dir que s'encén en el color blanc que consumeix 60mA de la intensitat màxima en 5V.

$$I_{\text{Max}} = 8\text{mòduls} \times 256\text{leds(Píxels)} / 1\text{mòdul} \times 60\text{mA} / 1\text{píxel} = 122,88\text{A} \quad (\text{Eq.1})$$

$$P_{\text{Total}} = 122,88\text{A} \times 5\text{V} = 614,4\text{W} \quad (\text{Eq.2})$$

5V 80A 400W



Figura 46. Font d'alimentació

Per satisfer els requeriments de voltatge, corrent i potència es trian la font d'alimentació: $I=160\text{A}$, $V=5\text{V}$, $P=800\text{W}$. Llavors, muntarem 2 fonts d'alimentacions de 80A en paral·leles.

6.2. Sistema de protecció elèctrica

Pel càlcul de protecció sap que la intensitat del circuit calculat és 2,67A. I seleccionat la intensitat nominal del circuit de protecció és 4A, amb una secció de 1,5mm² que té un corrent màxim admissible de 16A. Tots els càlculs es veuen a annex A Càlculs de protecció d'aquesta memòria.

6.3. Adaptador de nivell

Segon, el kit desenvolupament de NodeMCU en capítol 3 deia que els pins de l'operació és 3,3V per una altra paraula, el senyal de sortida de kit és 3,3V, l'alimentació i en canvi, el senyal de l'entrada dels mòduls LEDs RGB són 5V. Per tant, necessiten un convertidor o adaptador 3,3V a 5V viceversa.

6.3.1. Introducció Level shifer

Un adaptador de nivell (level shifter) és un component que permet convertir senyals lògiques de diferents nivells de tensió. És un dispositiu molt útil en un processador com Arduino.



Figura 47. Level shifter

Com saben que els processadors operen en diferents voltatges, sent els més habituals 5V, 3,3V. Tradicionalment els models d'Arduino més freqüents són de 5V. No obstant això, la majoria dels processadors moderns (Arduino Due, STM32, ESP8266, ESP32, Raspberry Pi) operen 3,3V.

Hi ha moltes formes de fer l'adaptació de nivells de tensió. Des solucions senzilles, com fer servir un divisor de tensió, a integrats específics que permeten altes velocitats de commutació en les dues direccions.

Un dels més habituals és emprar un simple transistor, que és una solució senzilla, d'alta velocitat, i bidireccional.

No obstant això, també hi ha mòduls comercials i integrats per a l'adaptació de nivell. Normalment és més senzill i barat utilitzar un Level Shifter comercial de construir el nostre a partir dels components electrònics. Aquests mòduls per a adaptació de nivell integren diferents números de canals bidireccional, generalment dos, quatre o vuit.

La velocitat de commutació d'aquests mòduls és prou ràpida per permetre usar-se en l'adaptació de tensió de senyals de comunicació, com a port sèrie (UART), I2C o SPI, sempre que disposem de canals suficients per a totes les línies de comunicació.

6.3.2. Esquema de muntatge

La connexió d'un Level Shifter (adaptador de nivell) és senzilla. En general tinença dos costats diferenciats, un per al dispositiu de més tensió (identificat amb HIGH) i un per al dispositiu de menor tensió (identificat amb LOW) que es veuran a la següent figura 46.

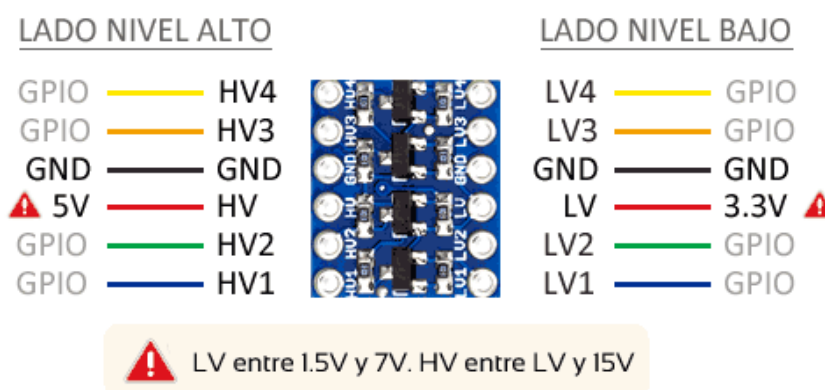


Figura 48. Esquema de muntatge

D'una banda, han de connectar les GND de tots dos dispositius a GND de la placa en els respectius costats de l'adaptador de final. GND ha de ser comú per a tots dos dispositius, per la qual cosa GND d'ambdós costats de la placa normalment estan units internament.

D'altra banda, han de proporcionar les dues tensions de referència de tots dos dispositius a l'Level Shifter. Per això tenim el pin HV, per la tensió Vcc del dispositiu de tensió més gran, i LV per Vcc del dispositiu de menor tensió.

Finalment, es poden usar-los tants pins GPIOs (canals) com necessiten. Els canals són bidireccionals, i funcionen tant per a senyals digitals, com analògiques, o sistemes de comunicació (UART, I2C, SPI).

7. MANUAL DE PROCEDIMENTS DE MARXA D'EQUIP

7.1. Muntatge físic del sistema

El sistema de la pantalla LEDs RGB remota per OTA consisteix en un dispositiu reben les informacions o senyals des d'un kit desenvolupament de NodeMCU 1.0 (ESP-12E) programat per un cable micro-USB o per OTA, i des d'allà envia-les a la pantalla dinàmica RGB que puguin observar per usuaris als quals proporcionar més efectes visuals dinàmics en colors més rics. El muntatge físic demostra en la figura següent:

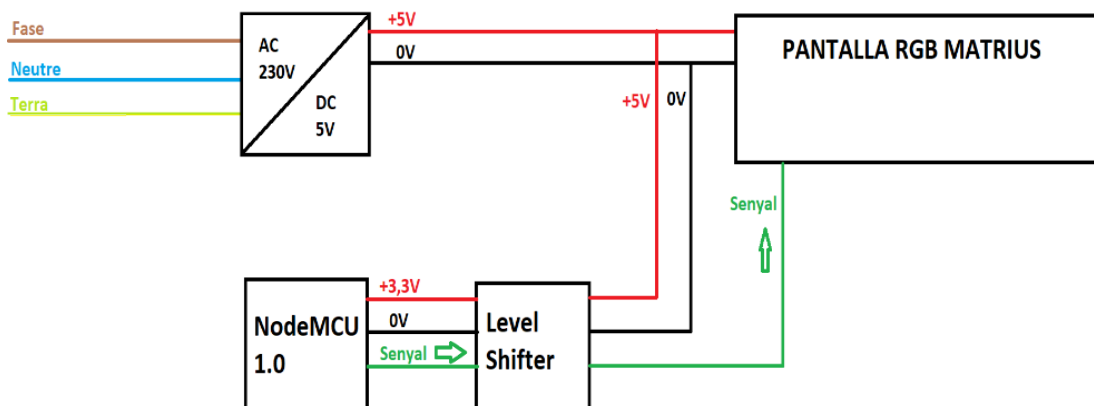


Figura 49. Esquema física

7.2. Connexió amb port serial

Primer de tot, ha de preparar l'ordinador que continguin el Arduino IDE instal·lat. Per qualsevol programa sempre fins i tot, a totes les llibreries que hi havia explicats pels capítols anteriors, que es mostren a la taula 6 son llibreries imprescindibles per les programacions:

Tots llibreries de NeoPixels
#include <Adafruit_GFX.h>
#include <Adafruit_NeoMatrix.h>
#include <Adafruit_NeoPixel.h>
Llibreries de ESP8266
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266mDNS.h>
#include <WiFiUdp.h>
#include <ArduinoOTA.h>

Taula 6. Llibreries imprescindibles

Després de compilar i descarregar-les al NodeMCU per un cable micro-USB, és impresentable per primer pas.

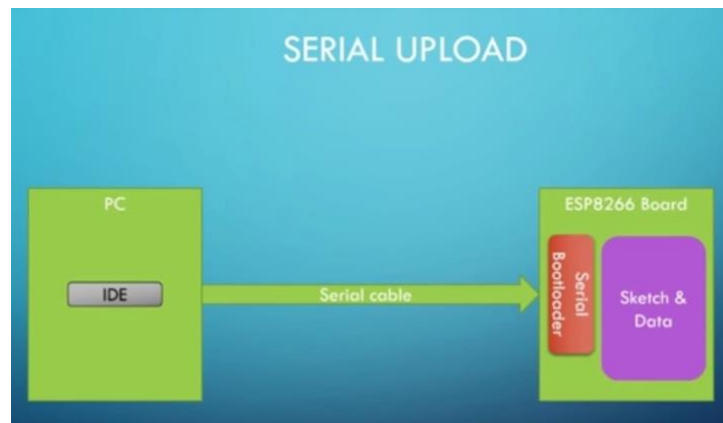


Figura 50. Càrrega el programa amb port sèrie

7.3. Actualització amb OTA

Encén el dispositiu de pantalla a l'endoll. Busca el port de WIFI a la seva xarxa, que poden trobar-lo al porto serial. I un cop connectar al port d'OTA, desconnectar el cable micro-USB, per descomptat que ja puguin actualitzar noves programacions per via OTA sense fil. Tot això, s'havia explicat en el capítol 5 d'aquesta memòria.



Figura 51. Càrrega amb OTA

8. RESUM DEL PRESSUPOST

L'import final per a la realització de la pantalla dinàmica de LEDs, 8 matrius de 8x32 LEDs RGB i es de dos mil vint-i-vuit euros amb quatre cèntims sense IVA.

9. CONCLUSIONS

El projecte compleix els objectius establerts, ja que s'ha pogut mostrar les lletres des d'un costat de la pantalla matrius LEDs RGB cap a l'altre, o de dalt a baix, vol dir que es poden controlar a quin sentit es mouen de les lletres, o les seves velocitats de moviment. També es poden canviar els colors de la font de les lletres, o de diferents de mides de les lletres. Gràcies per les llibreries de la "Adafruit" com "Neopixel", "NeoMatrix" i "GFX", són les tres llibreries principals per dissenyar a un matriu de la pantalla. I tots aquests espectaculars es rep els senyals digitals a través d'un kit desenvolupament de NodeMCU.

El NodeMCU és una petita placa que tenen un munt d'interfícies externes per facilitar les programacions. Dins el kit incorpora un mòdul xip ESP8266. És un xip té potents capacitats a bord de processament i emmagatzematge que li permeten integrar-se amb sensors i dispositius específics d'aplicació a través dels seus GPIOs amb un desenvolupament mínim i càrrega mínima durant el temps d'execució. En aquest projecte només s'ha utilitzat un GIOP de tipus digital per controlar la pantalla. A més, el mòdul ESP8266 en el qual ofereix una solució completa i autònoma de xarxes Wi-Fi, el que li permet allotjar l'aplicació o servir com a pont entre Internet i un microcontrolador de manera sense fil, que es diu l'actualització OTA.

L'actualització de OTA és el procés de carregar el microprogramari en el mòdul ESP8266 mitjançant una connexió Wi-Fi en lloc d'un port sèrie. Aquesta funcionalitat es torna extremadament útil en cas d'accés físic limitat o nul al mòdul, que faciliten a programar a la distància, a la manera sense fil.

Amb entorn d'Arduino IDE, que porten un munt de recursos de les llibreries obertes, un kit de desenvolupament NodeMCU incorpora WiFi que puguin aconseguir un munt d'aplicacions de control sense fil als dispositius, això és el futur de IOT. D'aquesta manera el projecte s'ha assolit de forma satisfactòria complint cada un dels punts que s'havien descrit a l'inici.

Wei Li

Graduat en enginyeria electrònica industrial i automàtica

Banyoles, 02 de Juny de 2019

10. RELACIÓ DE DOCUMENTS

El projecte consta de 5 documents. El primer document és la memòria seguit dels plànols, el plec de condicions, l'estat d'amidaments i el pressupost.

11. BIBLIOGRAFIA

Adafruit GFX Graphics Library. (<http://adafru.it/cBB>, 01 de febrer de 2019)

DC5V 32*8 pixels flexible led matrix Individually addressable LED Module RGB Digital Pixels Panel screen. (<https://www.aliexpress.com/w/wholesale-ws2812b-matrix.html>, 21 de gener de 2019)

Espressif ESP8266. (<https://www.espressif.com/en/products/hardware/esp8266ex/overview>, 19 de gener de 2019)

GitHub - esp8266/Arduino: ESP8266 core for Arduino. (<https://github.com/esp8266/Arduino>, 12 de febrer de 2019)

Malvino, Albert Paul. Principios de electrónica, Editorial McGraw-Hill, 6a ed. 2000.

NodeMCU, la popular placa de desarrollo con ESP8266. (<https://www.luisllamas.es/esp8266-nodemcu>, 10 de diciembre de 2018)

OTA Update, ESP8266 Arduino Core.

(http://esp8266.github.io/Arduino/versions/2.0.0/doc/ota_updates/ota_updates.html, 15 de febrer de 2019)

RGBmatrixPanel. (<http://adafru.it/aHj>, 09 de febrer de 2019)

Tutorial_32x16-32x32-rgb-led-matrix. (<https://learn.adafruit.com/32x16-32x32-rgb-led-matrix/overview>, 15 de gener de 2019)

12. GLOSSARI

GPIO: General Purpose Input/Output (Entrada/Sortida per a un Ús General)

IC: Integrated circuit (Circuit Integrat)

I²C: Inter-Integrated Circuit (Circuit Inter-Integrat)

IOT: Internet of Things (L'internet de les coses)

LED: Light Emitting Diode (díode emissor de llum)

MCU: Microcontroller Unit (Microcontrolador)

OTA: Over The Air (per sobre de l'aire)

RGB: Red, Green, Blue (colors primaris vermell, verd i blau)

SOC: System on chip (Sistema en xip)

SPI: Serial Peripheral Interface (Interfície per a perifèrics sèrie)

UART: Universal Asynchronous Receiver-Transmitter (Transmissor-receptor Asíncron Universal)

USB: Universal Serial Bus

A. CÀLCUL DE PROTECCIÓ

Per al càlcul del protecció s'haurà de complir:

$$I_b \leq I_n \leq I_z \quad (\text{Eq.3})$$

I_b la intensitat del circuit segons la previsió de càrregues.

I_n la intensitat nominal del circuit de protecció.

I_z la intensitat màxima admissible del conductor.

Segons el càlcul de potència total de la pantalla és 614,4W.

$$I_b = \frac{P_{\text{Total}}}{V \times \cos \theta} = \frac{614,4\text{W}}{230\text{V} \times 1} = 2,67\text{A} \quad (\text{Eq.4})$$

Llavors, se han seleccionat la intensitat nominal del circuit de protecció és 4A, amb una secció de 1,5mm² que té un corrent màxim admissible de 16A, i tot això compleix la equació 3, ja que:

$$I_b \leq I_n \leq I_z \Rightarrow 2,67\text{A} \leq 4\text{A} \leq 16\text{A} \quad (\text{Eq.5})$$

B. PROGRAMACIÓ

```
//LA PROGAMACIÓ DEL PROJECTE TFG

//Títol: Pantalla RGB dinàmica gestionada de manera remota

//Llibreria de Adafruit

#include <Adafruit_GFX.h>
#include <Adafruit_NeoMatrix.h>
#include <Adafruit_NeoPixel.h>
#include <Fonts/FreeMono9pt7b.h>
#include <Fonts/FreeSerif9pt7b.h>

//Llibreria de ESP8266
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266mDNS.h>
#include <WiFiUdp.h>
#include <ArduinoOTA.h>
#include <ESP8266WebServer.h>
#include <ESP8266HTTPUpdateServer.h>

const char* host = "esp8266-webupdate";

// Contrasenya de red
const char* ssid = "ssid";
const char* password = "password";
ESP8266WebServer server;
ESP8266WebServer httpServer(80);
ESP8266HTTPUpdateServer httpUpdater;

#ifndef PSTR
#define PSTR // Make Arduino Due happy
#endif

//Pin de senyal
#define PIN 12
```

```
//Velocitat a la qual es mou el text
#define VELOCIDAD 2000
#define VELOCIDAD_1 10
#define VELOCIDAD_2 200

//Declaració de la pantalla LEDs RGB
Adafruit_NeoMatrix matrix = Adafruit_NeoMatrix(32, 8, 4, 2, PIN,
    NEO_TILE_TOP + NEO_TILE_LEFT + NEO_TILE_ROWS + NEO_TILE_ZIGZAG +
    NEO_MATRIX_TOP + NEO_MATRIX_LEFT + NEO_MATRIX_COLUMNS + NEO_MATRIX_ZIGZAG,
    NEO_GRB + NEO_KHZ800);

//Definició de colors
const uint16_t colors[] = {
    matrix.Color(255,0,0), matrix.Color(0,255,0), matrix.Color(0,0, 255)};
uint16_t colores1 = matrix.Color(128, 0,128);
uint16_t colores2 = matrix.Color(128, 128,0);
uint16_t colores3 = matrix.Color(0, 128,128);
uint16_t colores4 = matrix.Color(0, 35,0);
uint16_t colores5 = matrix.Color(255, 215,215);
uint16_t colores6 = matrix.Color(255,192,203); //Pink
uint16_t colores7 = matrix.Color(128,0,128); //Purple
uint16_t colores8 = matrix.Color(0,255,255); //Cyan
uint16_t colores9 = matrix.Color(135,206,235); //SkyBlue
uint16_t colores10 = matrix.Color(127,255,170); //Auqamarin
uint16_t colores11 = matrix.Color(0,128,0); //Green
uint16_t colores12 = matrix.Color(255,255,0); //Yellow
uint16_t colores13 = matrix.Color(255,215,0); //Gold
uint16_t colores14 = matrix.Color(255,165,0); //Orange
uint16_t colores15 = matrix.Color(165,42,42); //Brown
uint16_t colores16 = matrix.Color(128,128,128); //Gray
uint16_t colores17 = matrix.Color(0,0,205); //MediumBlue

void setup() {
    //Paràmetres de ESP8266 con WIFI, OTA
    Serial.begin(115200);
```

```
Serial.println("Booting");

WiFi.mode(WIFI_STA);

WiFi.begin(ssid, password);

while (WiFi.waitForConnectResult() != WL_CONNECTED) {

    Serial.println("Connection Failed! Rebooting...");

    delay(5000);

    ESP.restart();

}

//Contrasenya d'OTA

    ArduinoOTA.setPassword("123");

//Preparació d'OTA

ArduinoOTA.onStart([]() {

    String type;

    if (ArduinoOTA.getCommand() == U_FLASH) {

        type = "sketch";

    } else {

        type = "filesystem";

    }

    Serial.println("Start updating " + type);

});

ArduinoOTA.onEnd([]() {

    Serial.println("\nEnd");

});

ArduinoOTA.onProgress([](unsigned int progress, unsigned int total) {

    Serial.printf("Progress: %u%%\r", (progress / (total / 100)));

});

ArduinoOTA.onError([](ota_error_t error) {

    Serial.printf("Error[%u]: ", error);

    if (error == OTA_AUTH_ERROR) {

        Serial.println("Auth Failed");

    } else if (error == OTA_BEGIN_ERROR) {
```

```
        Serial.println("Begin Failed");
    } else if (error == OTA_CONNECT_ERROR) {
        Serial.println("Connect Failed");
    } else if (error == OTA_RECEIVE_ERROR) {
        Serial.println("Receive Failed");
    } else if (error == OTA_END_ERROR) {
        Serial.println("End Failed");
    }
});

ArduinoOTA.begin();

Serial.println("Ready v2.0");

Serial.print("IP address: ");

Serial.println(WiFi.localIP());

server.on("/restart", []() {
    server.send(200, "text/plain", "Restarting...");
    delay(1000);
    ESP.restart();
});

server.begin();

matrix.begin();

matrix.setTextWrap(false);

matrix.setBrightness(40);

matrix.setTextColor(colors[0]);
}

int x    = matrix.width();

int y = matrix.height();

int pass = 0;

int i=1;

int estado = 1;

void loop() {
```

```
ArduinoOTA.handle();

server.handleClient();

matrix.fillScreen(0);

switch (estado)
{
  case 1: //Informació principal
    matrix.setTextSize(1);
    matrix.setFont();
    matrix.setTextColor(colors[0]);
    matrix.setCursor(x, 0);
    matrix.println(F("CHAMPIONS LEAGUE"));
    matrix.setCursor(x, 8);
    matrix.println(F("SEMIFINALES (IDA)"));

    if(--x < 16 )
    {
      x=matrix.width();
      estado=2;
    }

    matrix.show();
    delay(VELOCIDAD_1);
    break;

  case 2: //Informació repetició
    for(i=0; i<=2; i++)
    {
      matrix.setTextColor(colors[i]);
      matrix.setCursor(16, 0); //
      matrix.println(F("CHAMPIONS LEAGUE"));
      matrix.setCursor(16, 8);
      matrix.println(F("SEMIFINALES (IDA)"));
      matrix.show();
    }
  }
}
```

```
        delay(VELOCIDAD);
    }
    y=15;
    x=matrix.width();
    estado=3;
    break;

case 3: // Des de baix puja a dalt
    matrix.fillScreen(colores4);
    matrix.setTextColor(colores1);
    matrix.setCursor(7, y);
    matrix.println(F("BARCELONA-LIVERPOOL"));
    matrix.setCursor(7, y+8);
    matrix.println(F("MARTES, 30/ABR 21H"));

    if(--y < 0 )
    {
        y=-15;
        estado=4;
        i=1;
    }
    matrix.show();
    delay(VELOCIDAD_2);
    break;

case 4: //
    matrix.fillScreen(colores4);
    matrix.setTextColor(colores1);
    matrix.setCursor(7, 0);
    matrix.println(F("BARCELONA-LIVERPOOL"));
    matrix.setCursor(7, 8);
```



```
        matrix.println(F("MARTES,30/ABR 21H"));

        matrix.show();

        delay(5000);

        estado=5;

        break;

case 5: //De dalt a baix

        matrix.fillScreen(10);

        matrix.setTextColor(colores2);

        matrix.setCursor(16, y);

        matrix.println(F("BARCELONA-GIRONA"));

        matrix.setCursor(7, y+8);

        matrix.println(F("AMISTOSO,01/JUN 20H"));

        if(++y > 0 )

        {

            y=15;

            estado=6;

            i=0;

        }

        matrix.show();

        delay(VELOCIDAD_2);

        break;

case 6: //

        matrix.fillScreen(10);

        matrix.setTextColor(colores2);

        matrix.setCursor(16, y);

        matrix.println(F("BARCELONA-GIRONA"));

        matrix.setCursor(7, y+8);

        matrix.println(F("AMISTOSO,01/JUN 20H"));

        delay(2000);

        i=0;

        y=15;
```

```
x=matrix.width();  
estado=7;  
break;  
  
case 7: //Cada lletra té el seu color  
matrix.setCursor(25, 4);  
matrix.setTextColor(colores1);  
matrix.print(F("M"));  
matrix.setTextColor(colores2);  
matrix.print(F("E"));  
matrix.setTextColor(colores3);  
matrix.print(F("N"));  
matrix.setTextColor(colores5);  
matrix.print(F("U "));  
matrix.setTextColor(colores6);  
matrix.print(F("Y "));  
matrix.setTextColor(colores7);  
matrix.print(F("P"));  
matrix.setTextColor(colores8);  
matrix.print(F("O"));  
matrix.setTextColor(colores9);  
matrix.print(F("S"));  
matrix.setTextColor(colores10);  
matrix.print(F("T"));  
matrix.setTextColor(colores11);  
matrix.print(F("R"));  
matrix.setTextColor(colores12);  
matrix.print(F("E"));  
matrix.setTextColor(colores13);  
matrix.print(F("S"));  
matrix.show();  
delay(3000);
```

```
        i=0;
        x=matrix.width();
        estado=8;
    break;

case 8: //Canvia la mida de lletra
    matrix.setTextSize(2);
    matrix.setCursor(0, 0);
    matrix.setTextColor(colores1);
    matrix.print(F("M"));
    matrix.setTextColor(colores2);
    matrix.print(F("E"));
    matrix.setTextColor(colores3);
    matrix.print(F("N"));
    matrix.setTextColor(colores5);
    matrix.print(F("U"));
    matrix.setTextColor(colores6);
    matrix.print(F("Y"));
    matrix.setTextColor(colores7);
    matrix.print(F("P"));
    matrix.setTextColor(colores8);
    matrix.print(F("O"));
    matrix.setTextColor(colores9);
    matrix.print(F("S"));
    matrix.setTextColor(colores10);
    matrix.print(F("T"));
    matrix.setTextColor(colores11);
    matrix.print(F("R"));
    matrix.setTextColor(colores12);
    matrix.print(F("E"));
    matrix.setTextColor(colores13);
```

```
matrix.print(F("S"));
matrix.show();
delay(3000);
x=matrix.width();
i=0;
estado=9;
break;

case 9: //Canvia la font de lletra
matrix.setTextSize(1);
matrix.setTextColor(colores17);
matrix.setFont(&FreeSerif9pt7b);
matrix.setCursor(x, 12);
matrix.println(F("Universitat de Girona"));
matrix.show();
    if(--x < -32 ) //(128-126)/2=1
    {
        x=matrix.width();
        i++;
        if(i>2)
        {
            estado=1;
        }
    }

matrix.show();
delay(VELOCIDAD_1);
break;
}

}
```