

Tutoriale KidsBlock

1. Descărcarea codului și a bibliotecii

Descărcați link-uri pentru fișierele de cod ale tutorialului:

📄 Click to download

Name	Date modified
hello Keystudio	4/18/2024 1:14 PM
servo	4/18/2024 1:15 PM
1.1Blink.sb3	1/31/2024 1:57 PM
1.2PWM.sb3	4/23/2025 8:20 AM
1.3Button.sb3	1/31/2024 1:57 PM
1.4Self-Locking-Button.sb3	1/31/2024 1:57 PM
1.5Lighting-System.sb3	1/31/2024 1:57 PM
2.1Photocell-sensor.sb3	1/31/2024 1:57 PM
2.2Light-Control-System.sb3	1/31/2024 1:57 PM
3.1PIR-Motion-Sensor.sb3	4/23/2025 8:33 AM
3.2Passive-Buzzer.sb3	4/23/2025 8:35 AM
3.3Buzzer-Tone.sb3	1/31/2024 1:57 PM
3.4Buzzer-Music.sb3	1/31/2024 1:57 PM
3.5Alarm-System.sb3	4/23/2025 8:44 AM
4.1Steam Sensor.sb3	1/31/2024 1:57 PM

2. Instalarea software-ului

2.1 Instalarea KidsBlock

2.1.1 Instalați Kidsblock pe sistemul Windows

1. Puteți descărca KidsBlock de la:

- Legătură: <http://xiazai.keyesrobot.cn/KidsBlock.exe>

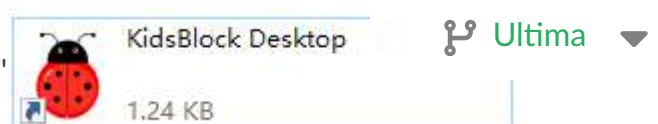
Scratch > 1.Installaion guidance > Software

• Fișier furnizat:

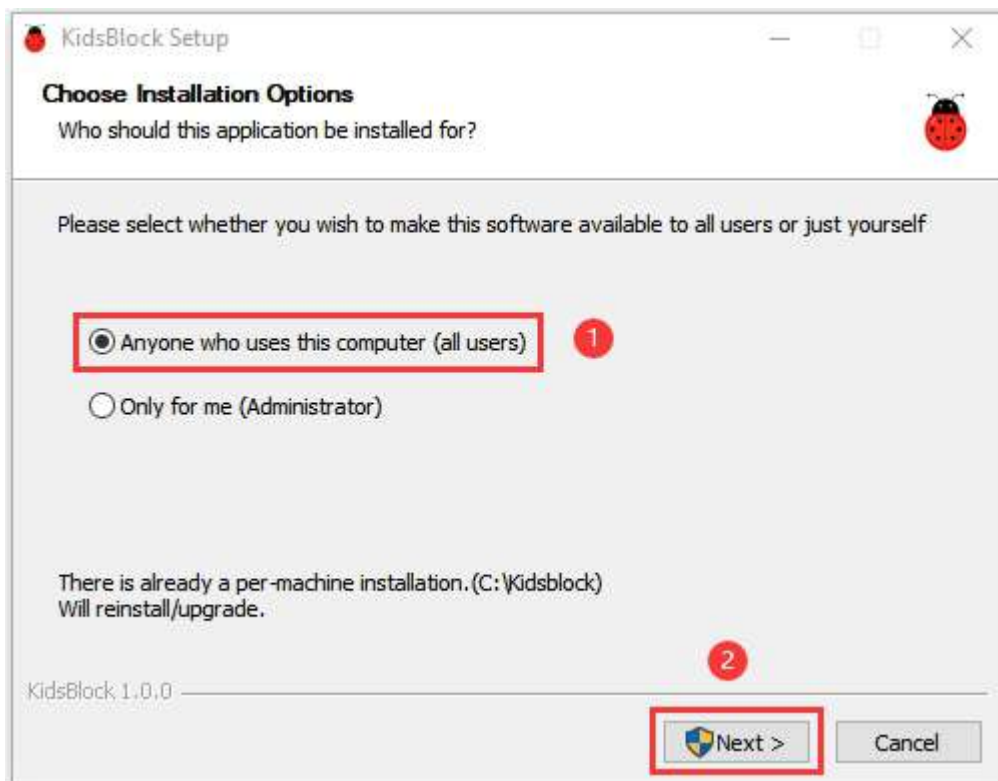


KidsBlock.exe
KidsBlock as a self-contained ...
Kidsblock.cc Team

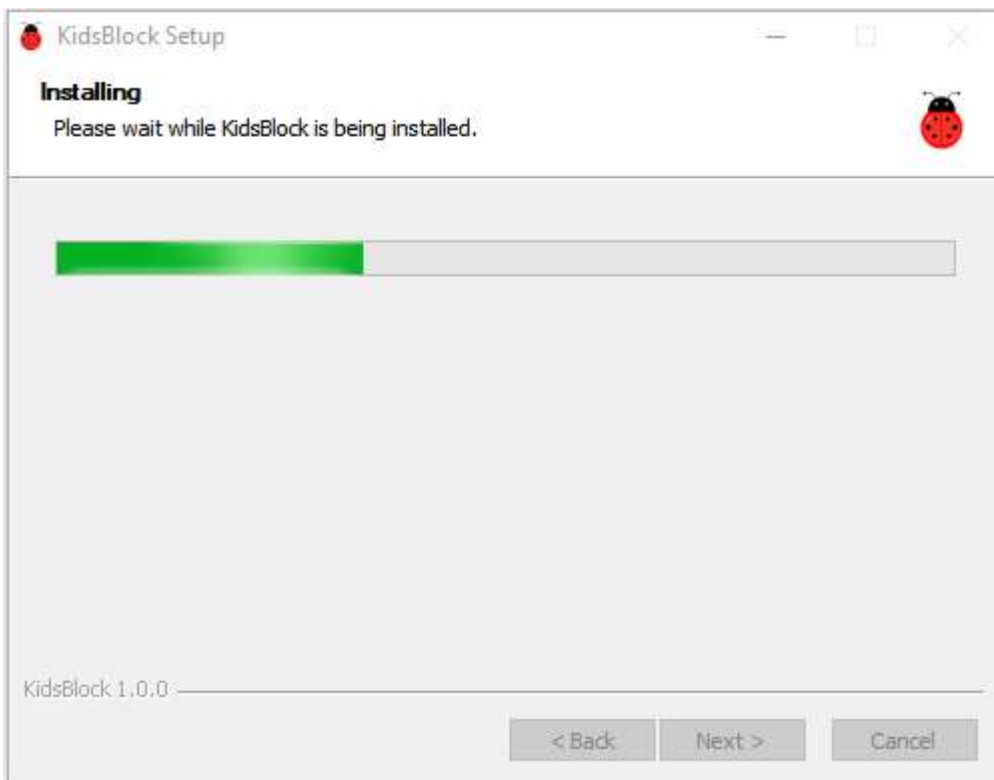
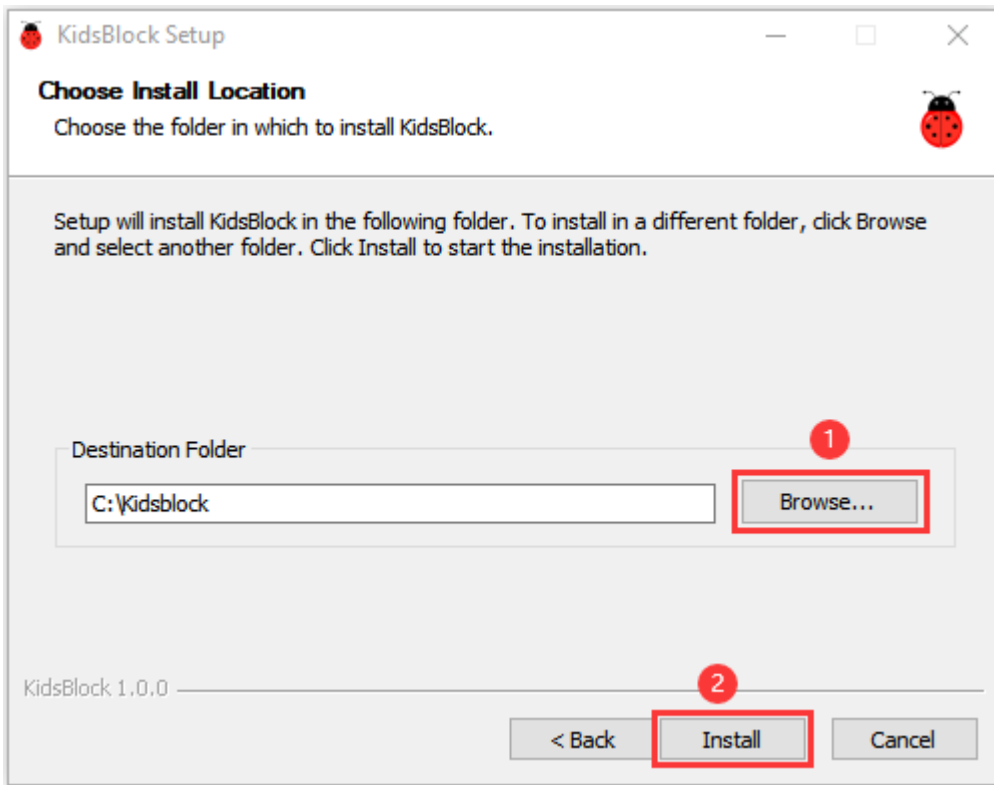
2. După descărcare, faceți clic pe "KidsBlock.exe"



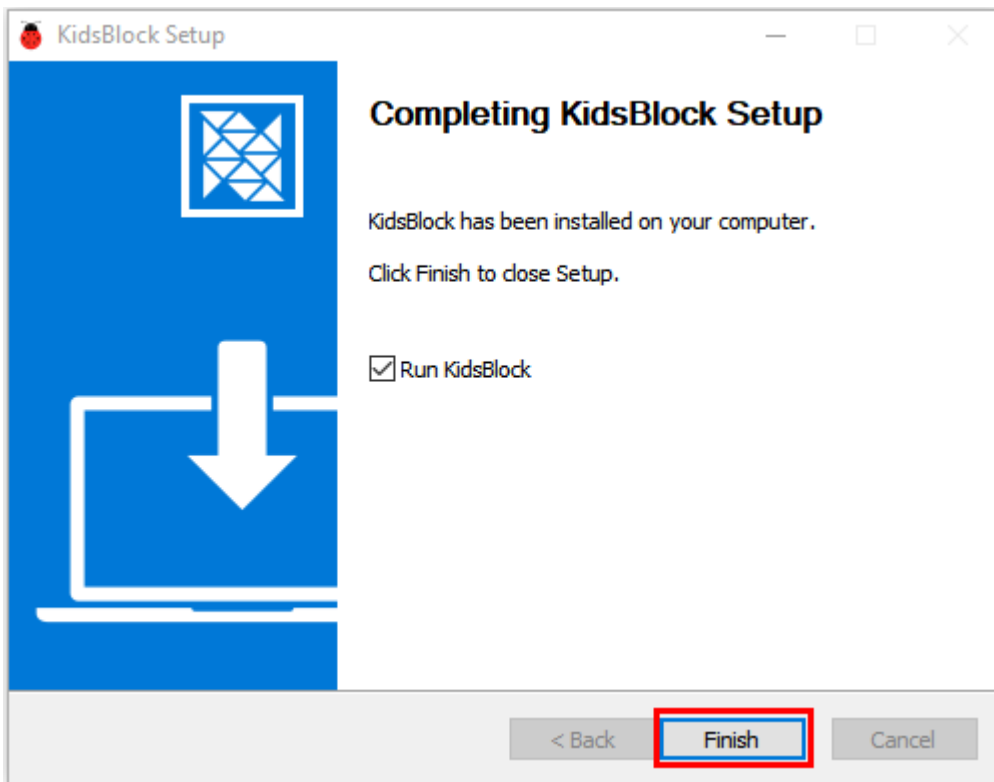
3. Bifați "Oricine folosește acest computer (toți utilizatorii)" și apoi faceți clic pe "Următorul".



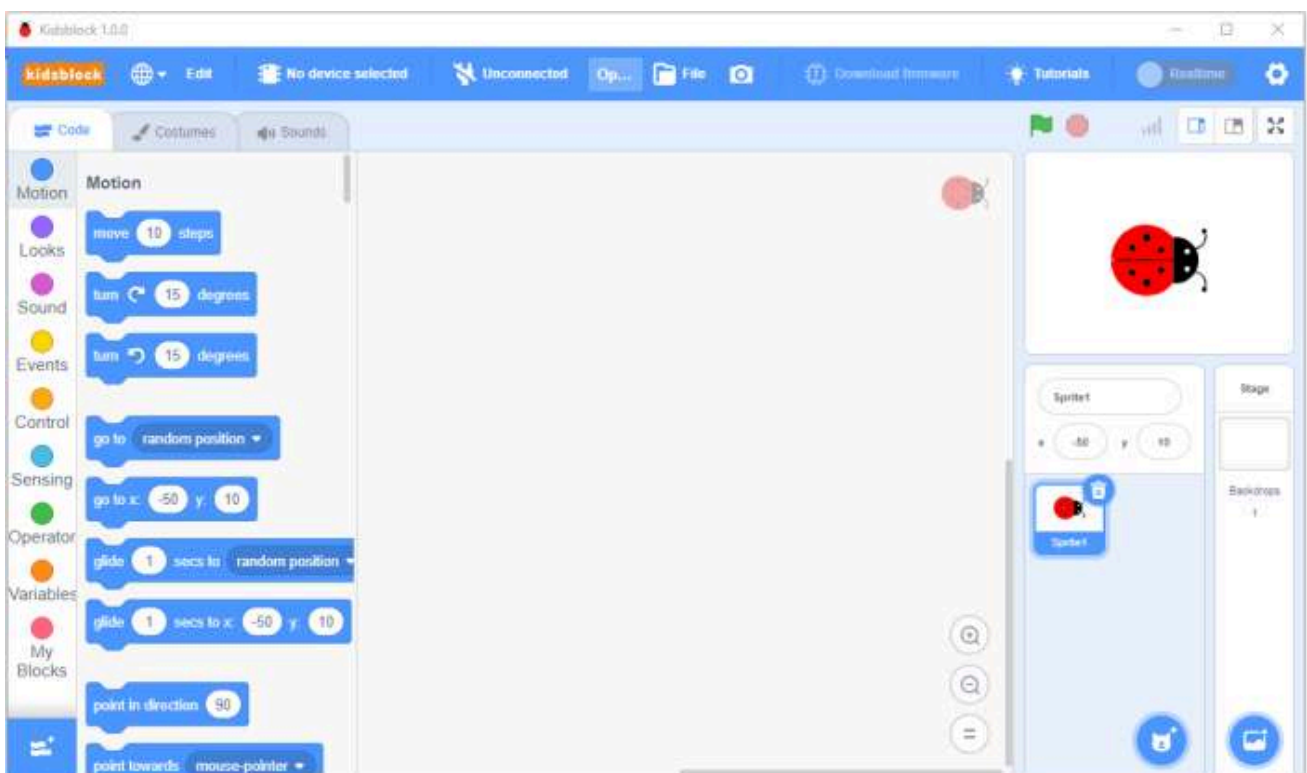
4. Faceți clic pe "Răsfoiți..." pentru a alege o cale de instalare (Aici alegem Discul C; puteți selecta oriunde doriți) și faceți clic pe "Instalare". Acum se instalează!



5. După finalizarea instalării, faceți clic pe "Finalizare" pentru a-l deschide.



6. Dacă apare un avertisment, vă rugăm să faceți clic pe "**Permiteți accesul**" pentru a intrați în pagina principală a software-ului.



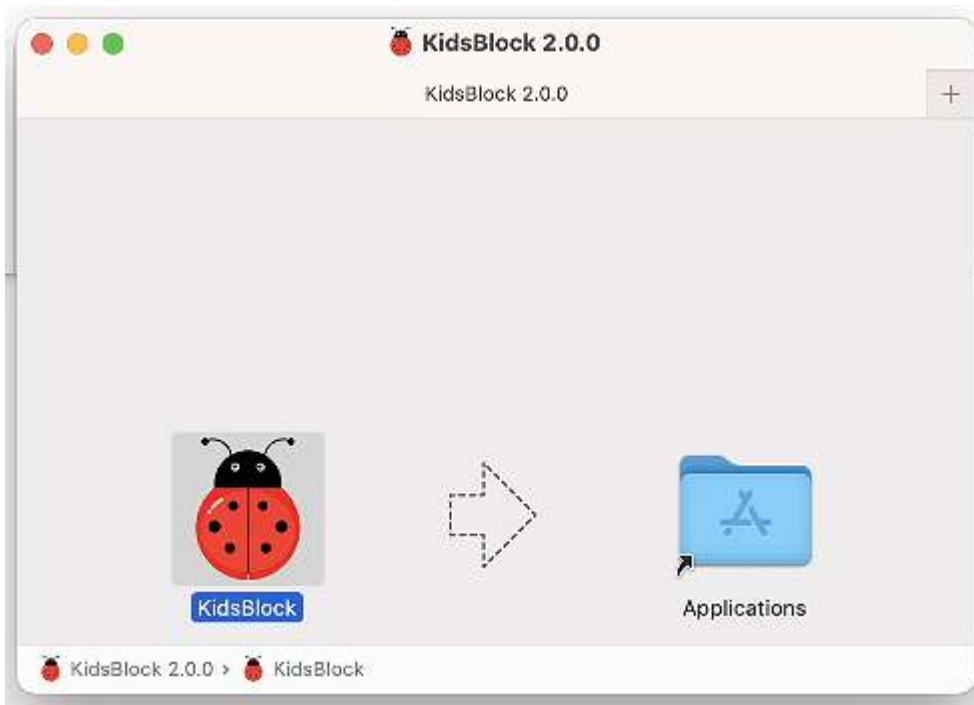
2.1.2 Instalați Kidsblock pe MacOS

1. Vă rugăm să descărcați mai întâi pachetul Kidsblock:
<http://xiazai.keyesrobot.cn/KidsBlock.dmg>



KidsBlock 2.0.0.dmg

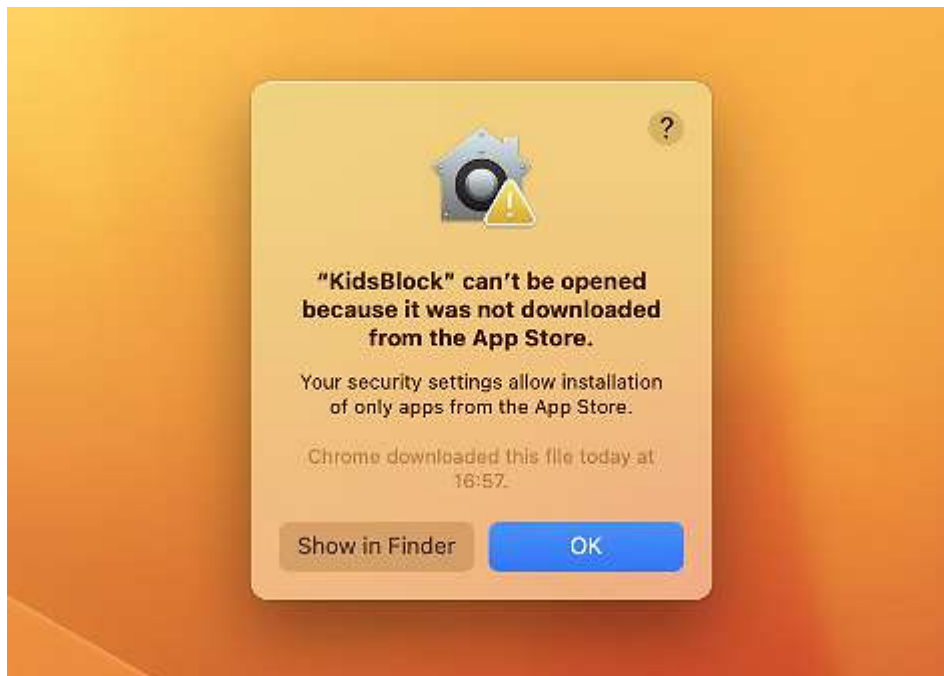
2. Faceți clic pe KidsBlock și trageți "**KidsBlock Desktop**" în "**Aplicații**", așa cum se arată mai jos.



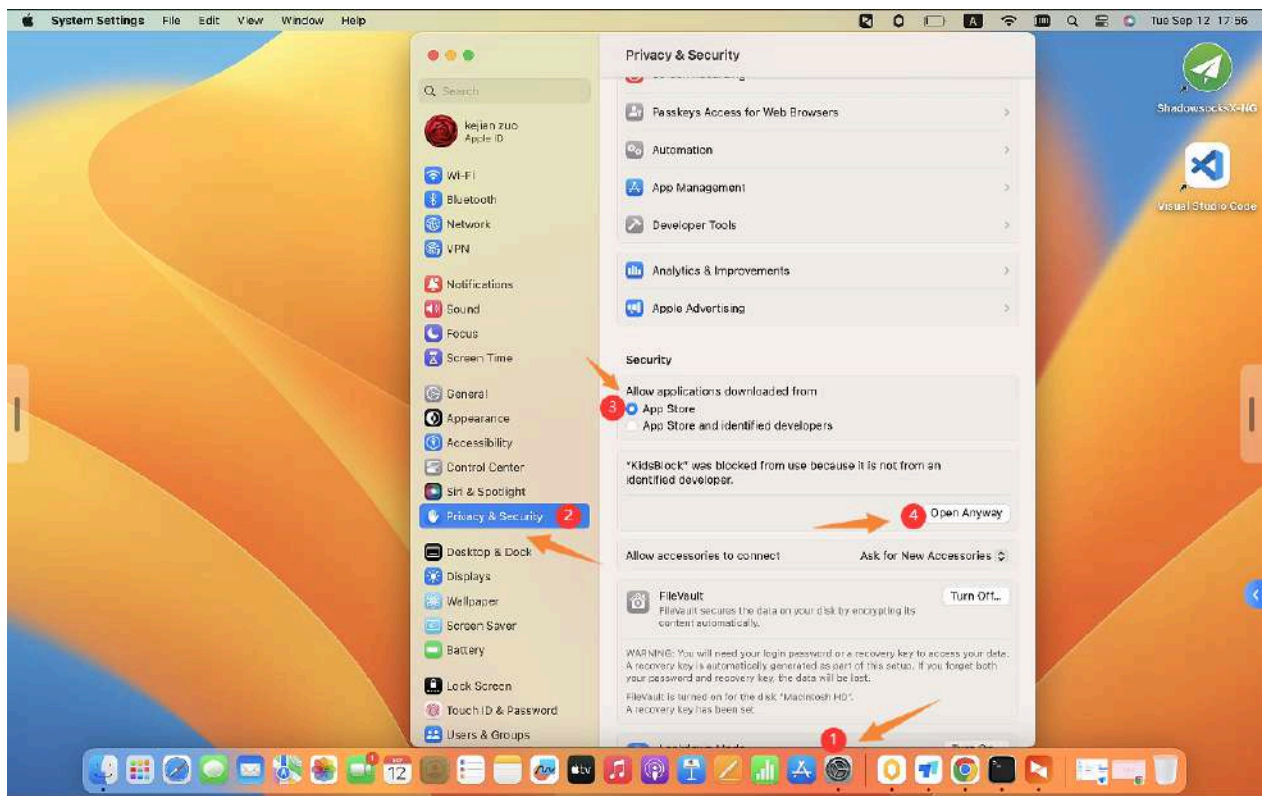
3. După instalare, pictograma KidsBlock va apărea în panoul de operare:



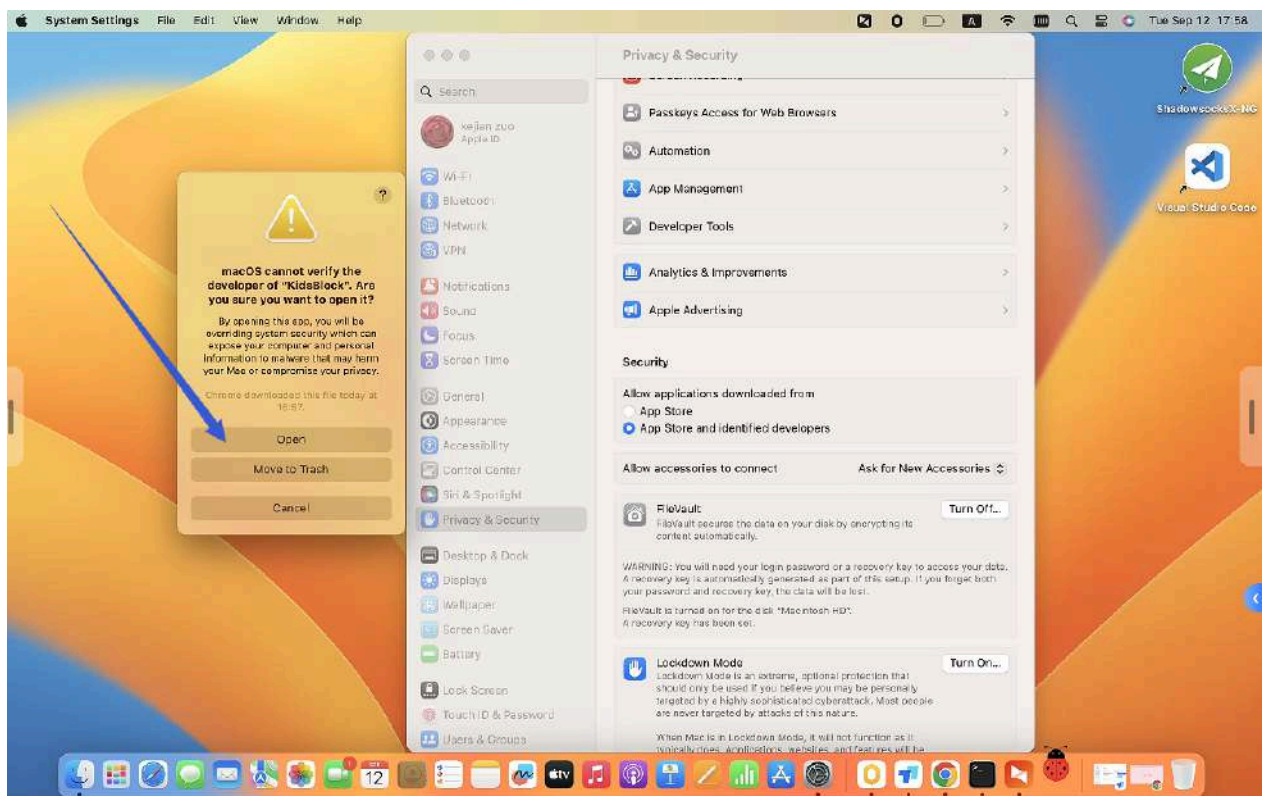
4. Faceți clic pe pictograma KidsBlock pentru a intra în software. Dacă nu reușește, vă rog modificați unele setări ale computerului pentru a le reintroduce. Acest lucru se datorează faptului că Mac Sistemele, în mod implicit, autorizează instalarea doar în App Store, deci altele nu pot fi instalate.



5. Deschideți setările și faceți clic pe Confidențialitate și securitate. Comutați opțiunea de securitate la "App Store și dezvoltatori aprobați" și faceți clic pe "Încă deschis".



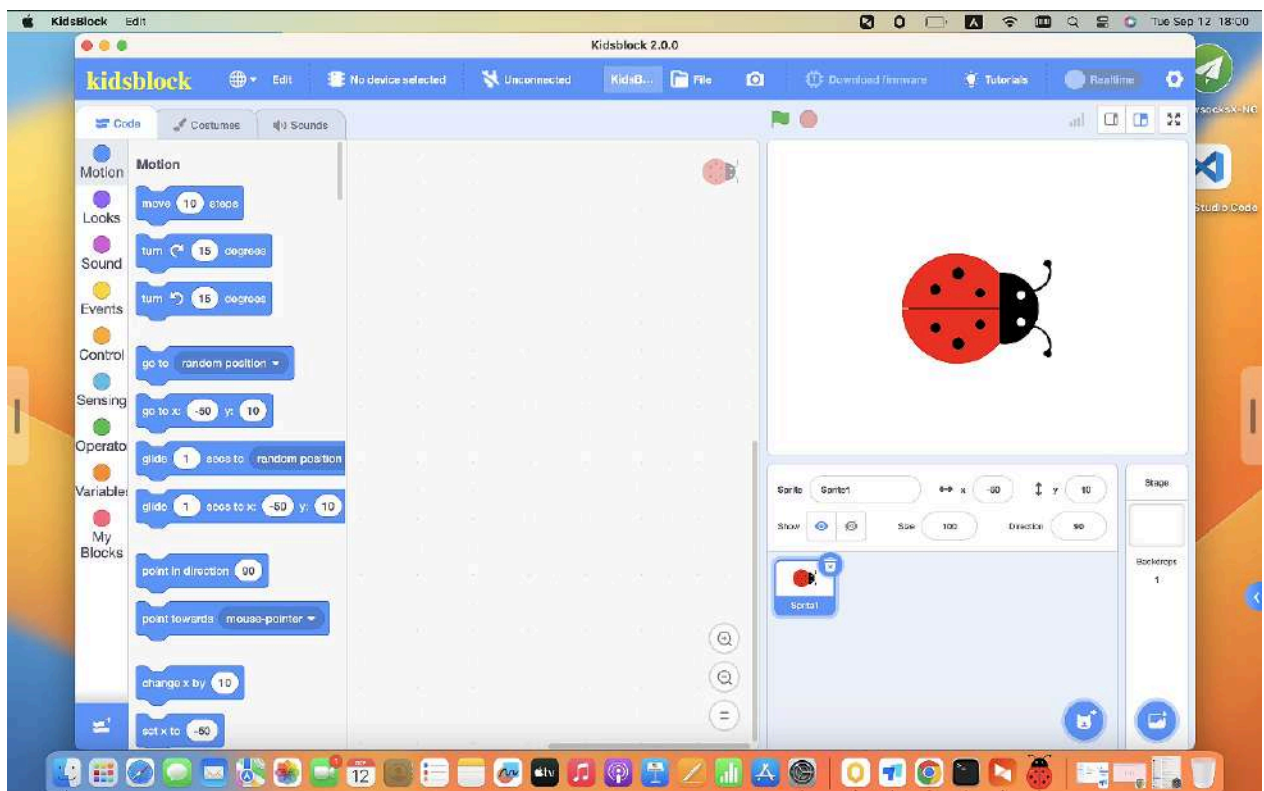
6. Faceți clic pe "Deschidere" pentru a reintroduce software-ul blocat.



7. După setări, poate funcționa normal.



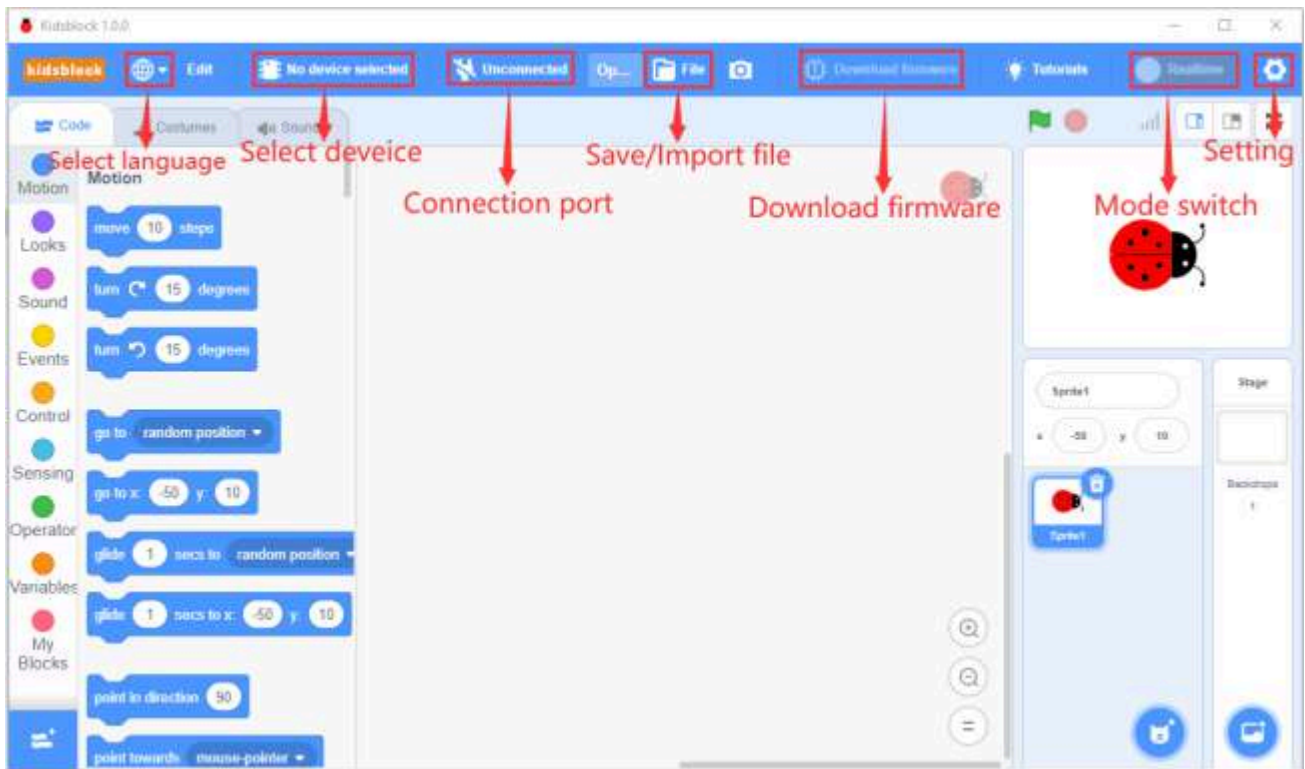
8. Interfața de pornire este ca mai jos. Acum bucură-te de călătoria ta de programare!




2.2 Ghid software

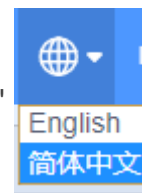
(Următoarele demonstrații se bazează pe sistemul Windows și sunt doar un referință pentru MacOS.)

2.2.1 Distribuția funcției paginii principale 1



2.2.2 Selectați limba

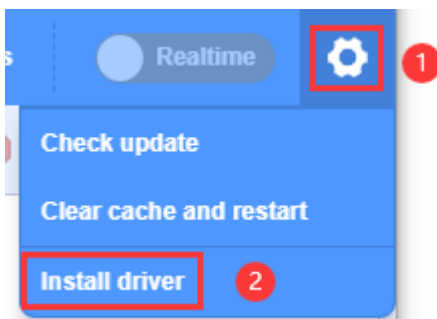
Faceți clic  pentru a selecta "Engleză" sau "简体中文"



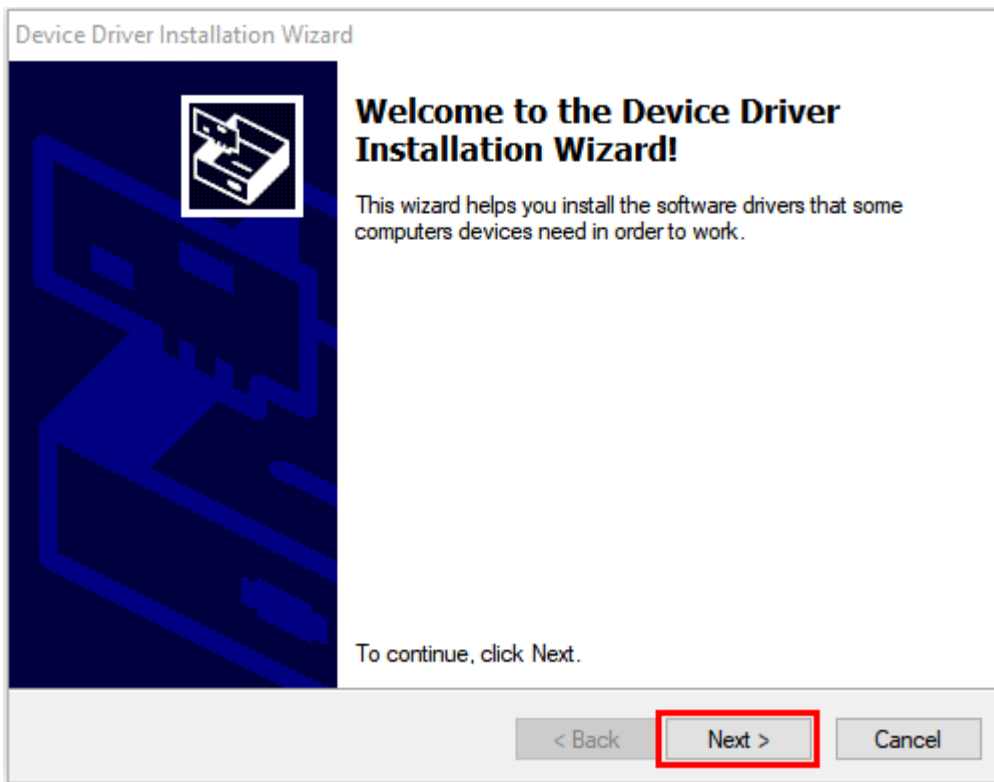
2.2.3 Instalați driverul

NOTĂ: Dacă computerul dumneavoastră a instalat deja un driver, vă rugăm să Săriți peste această secțiune. Dacă nu, vă rugăm să urmați acești pași.

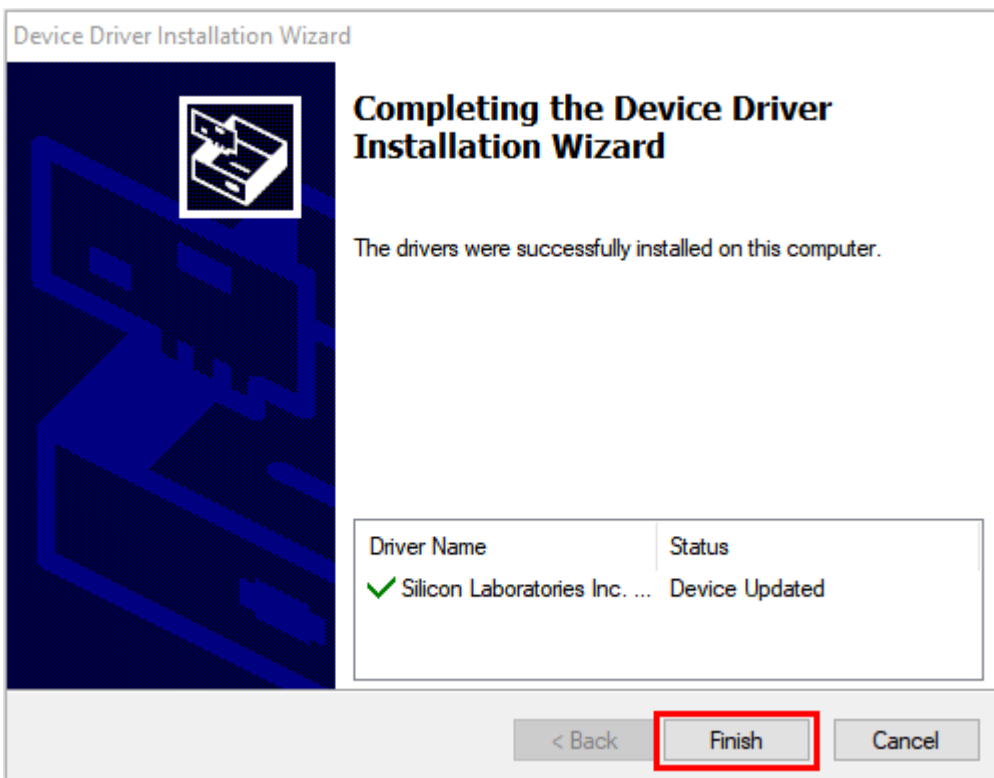
- Faceți clic  pentru a selecta "Instalare driver".



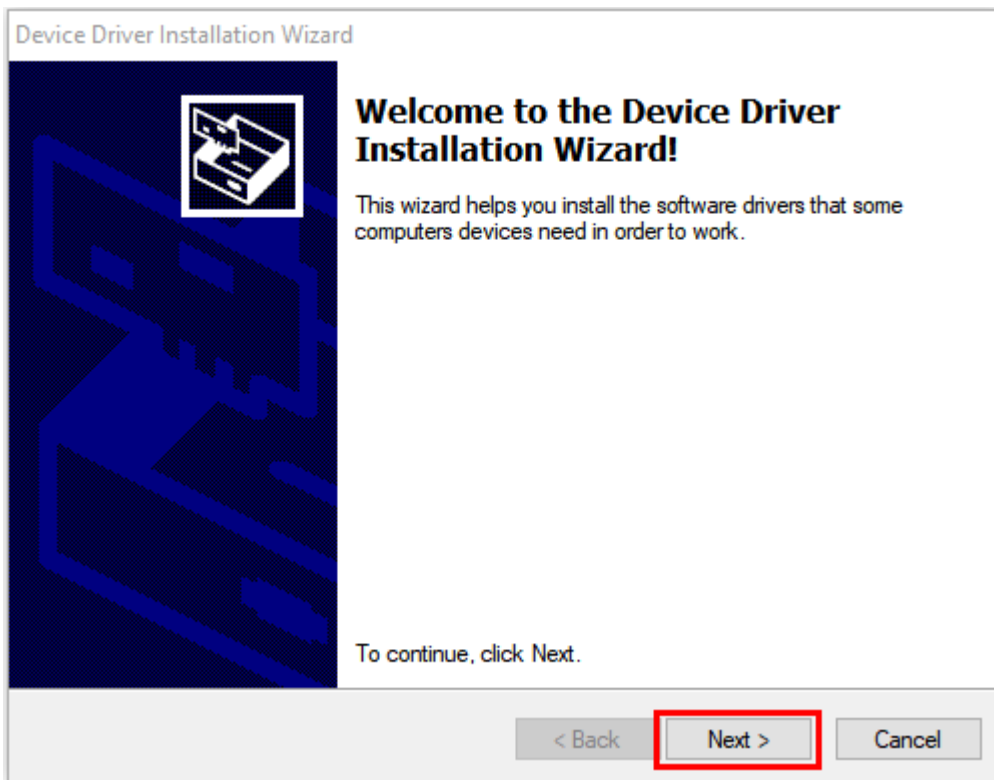
- Faceți clic pe "**Următorul**" când vedeți instalarea driverului de dispozitiv magician:



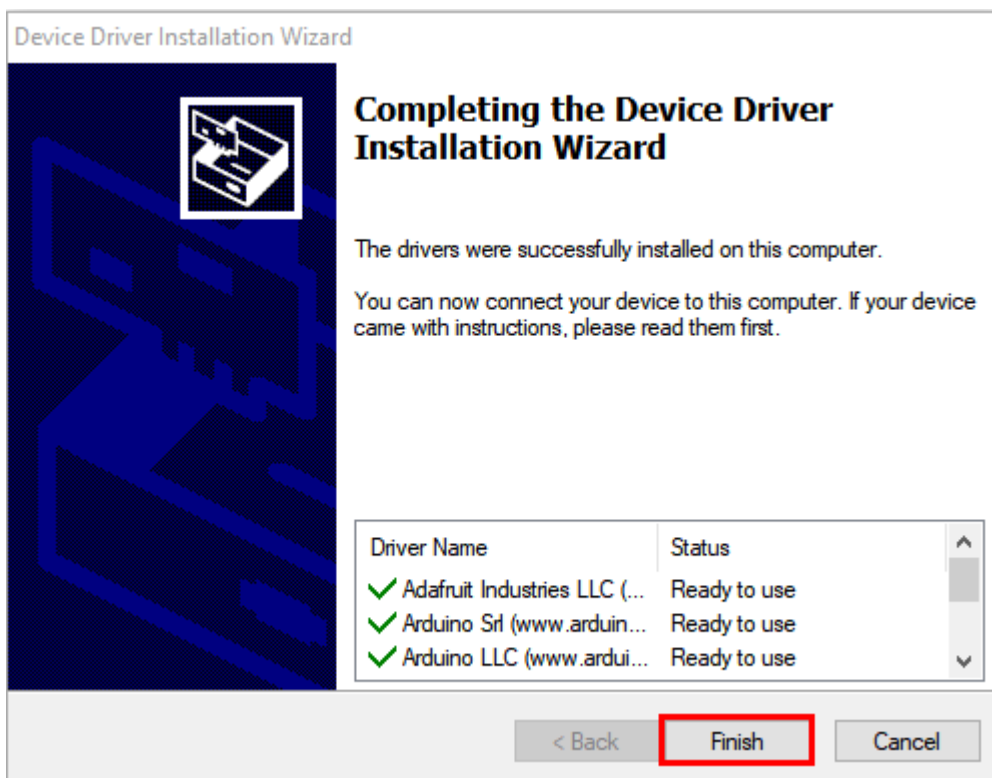
- Faceți clic pe "Finalizare".



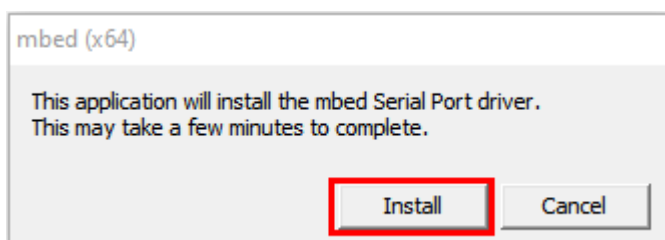
- Faceți clic pe "Următorul".



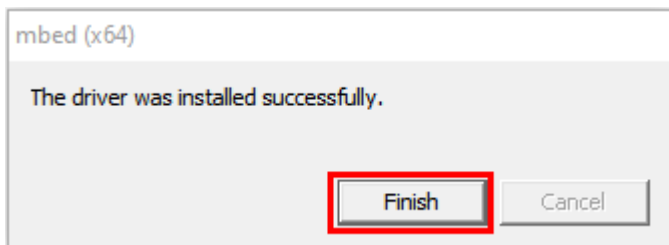
- Faceți clic pe "Finalizare".



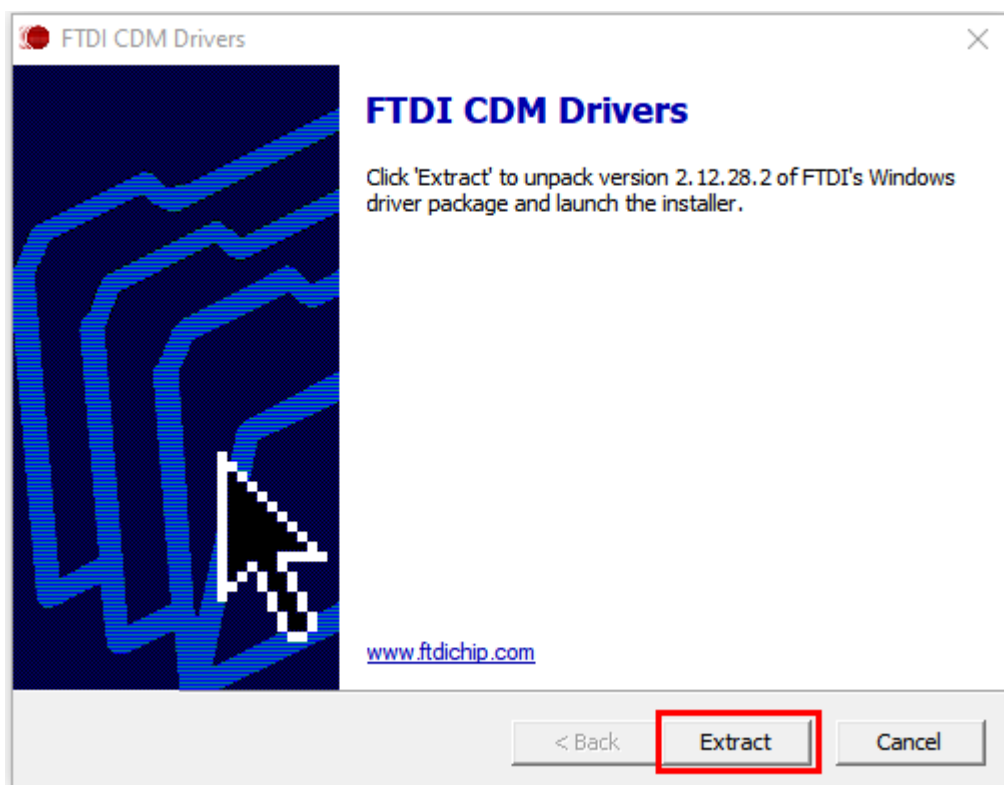
- Dacă apare un avertisment, faceți clic pe "Permite". Apoi faceți clic pe "Instalare".



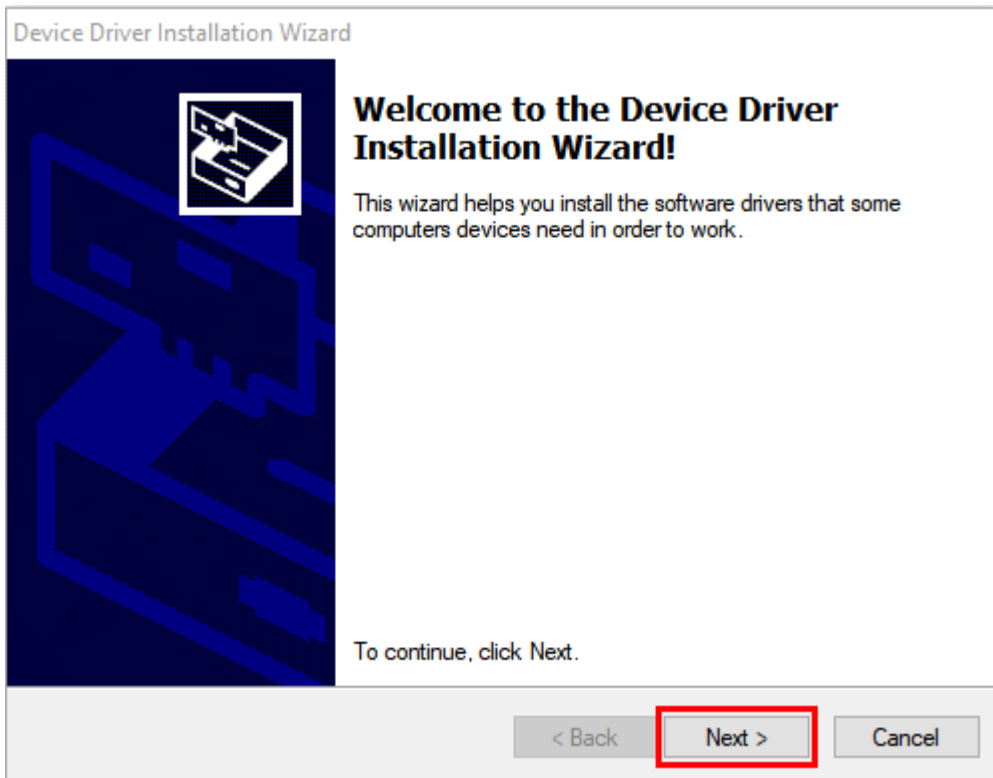
- Atingeți "Finalizare".



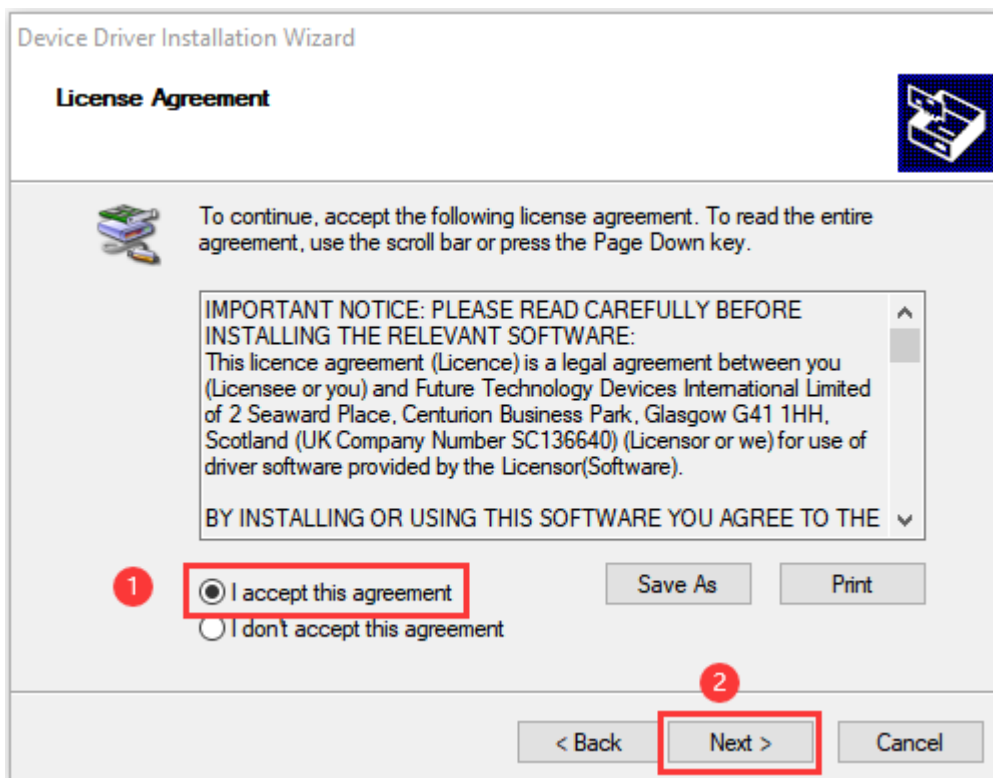
- Faceți clic pe "Extrageți".



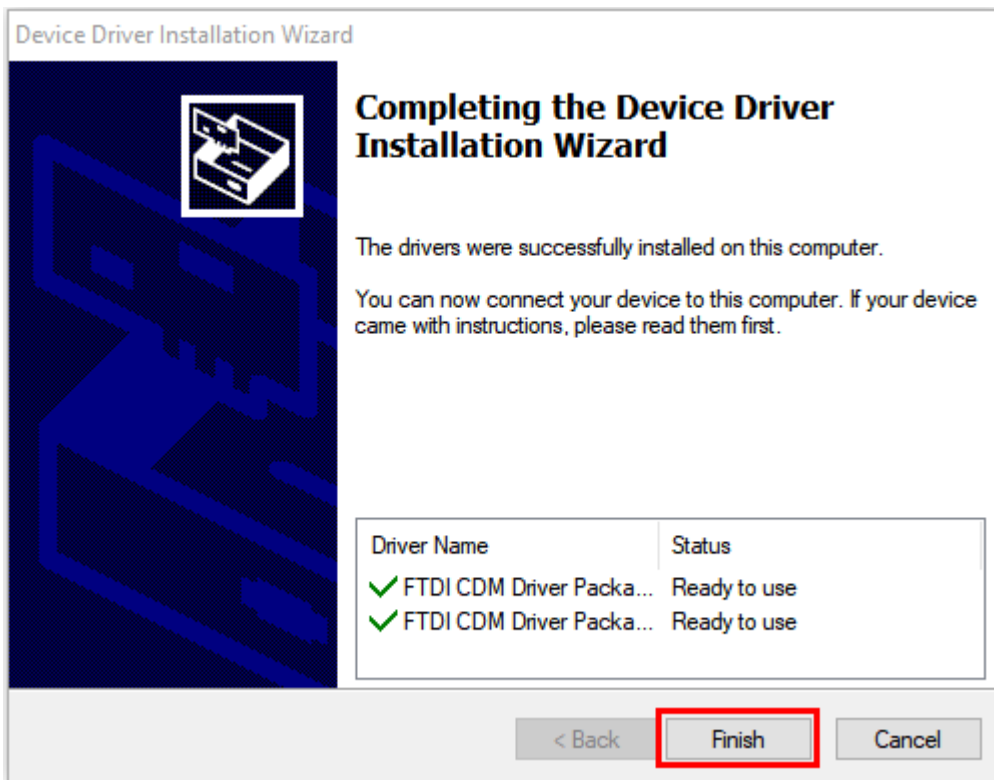
- Faceți clic pe "Următorul".



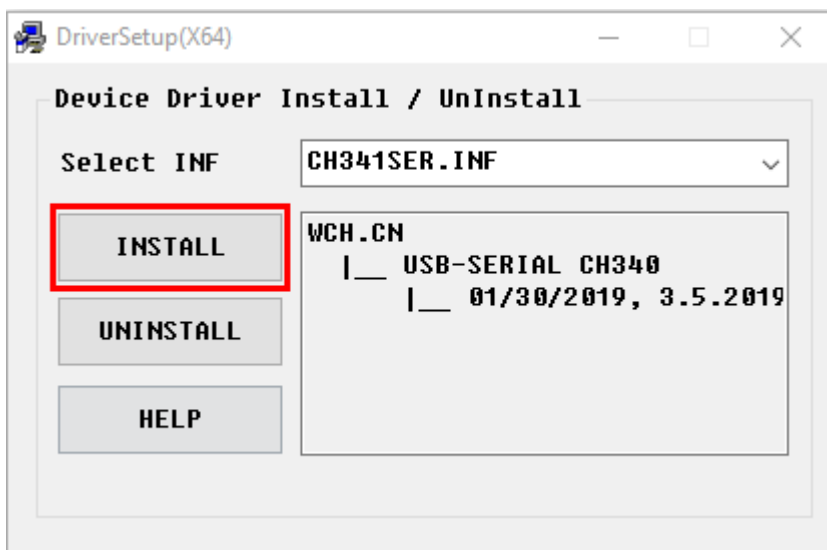
- Bifați "Accept acest acord" și faceți clic pe "Următorul".



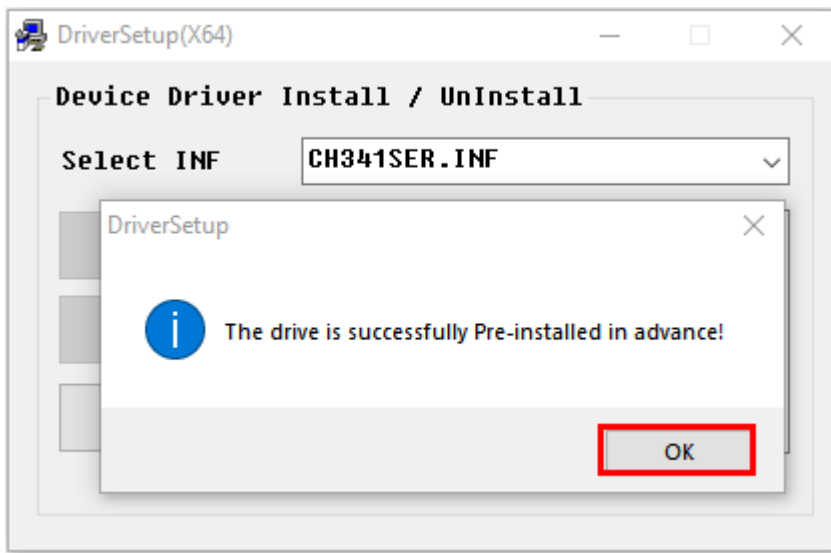
- Faceți clic pe "Finalizare".



- Alegeți "INSTALARE".




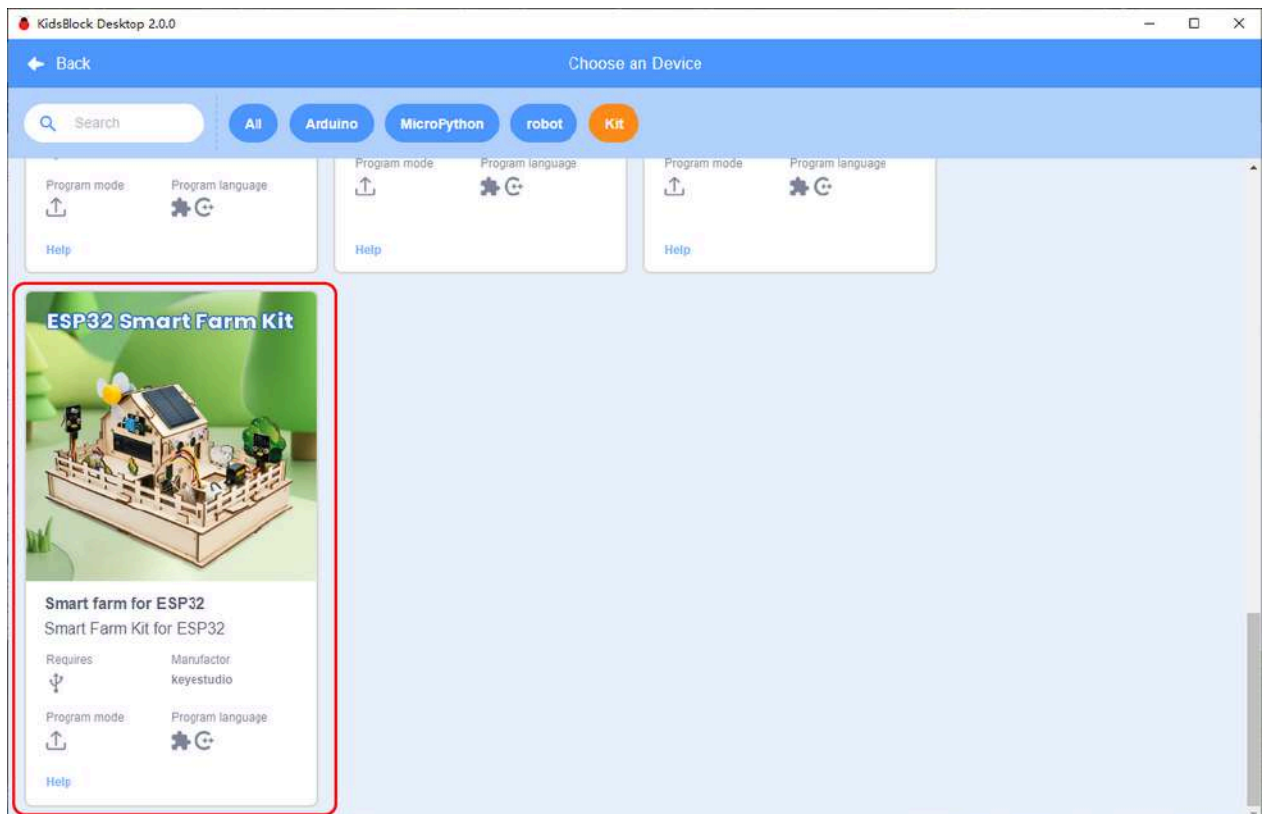
- Câteva secunde mai târziu, driverul va fi instalat cu succes. Apoi faceți clic pe "OK".



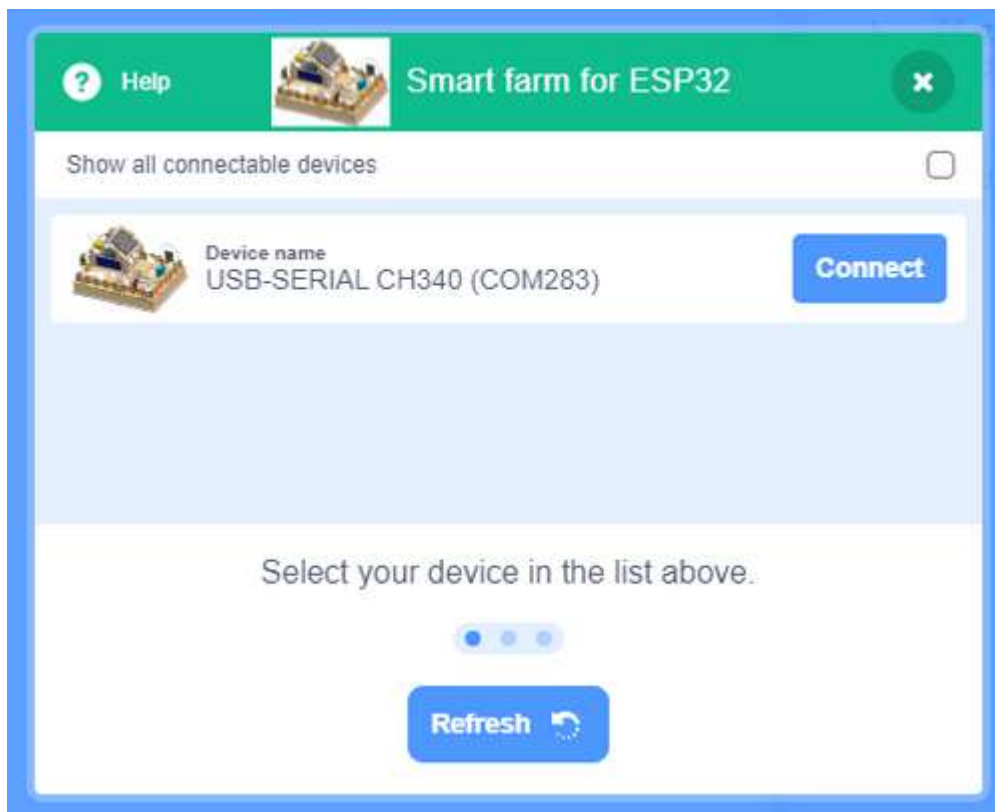
2.2.4 Selectați dispozitivele

Alegeți un dispozitiv și un port serial

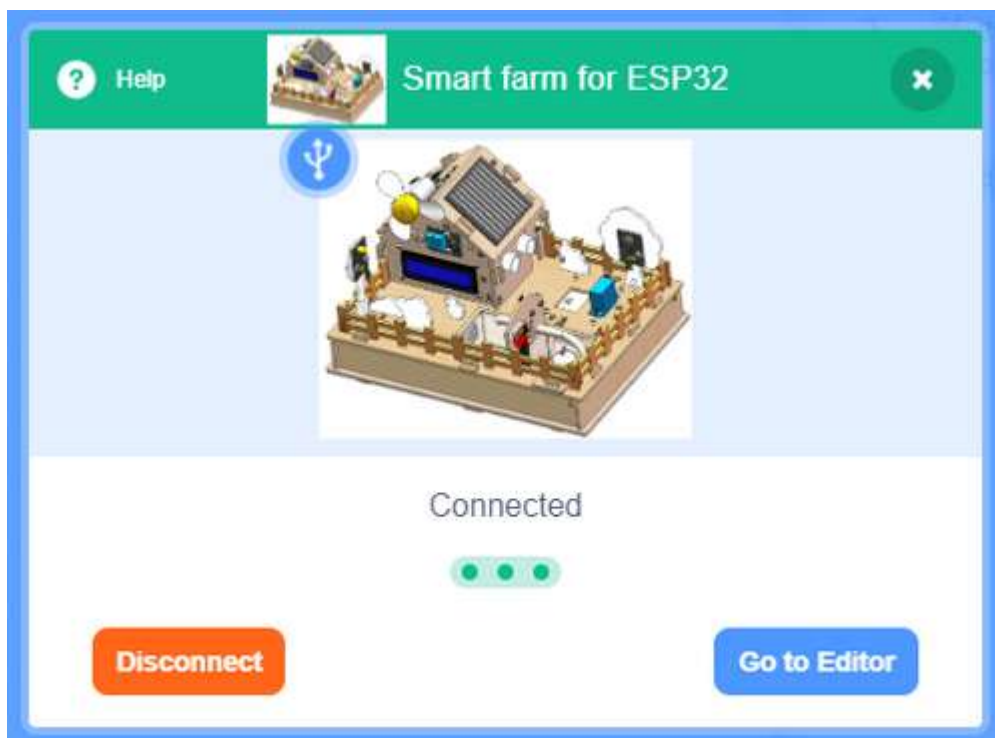
- Faceți clic  **No device selected** pentru a alege un dispozitiv.
- Aici intrăm în **Kit** pentru a găsi **ferma inteligentă pentru ESP32** și a o adăuga. Rețineți că toți senzorii sunt incluși în acest kit, deci nu este nevoie să importați-le suplimentar.



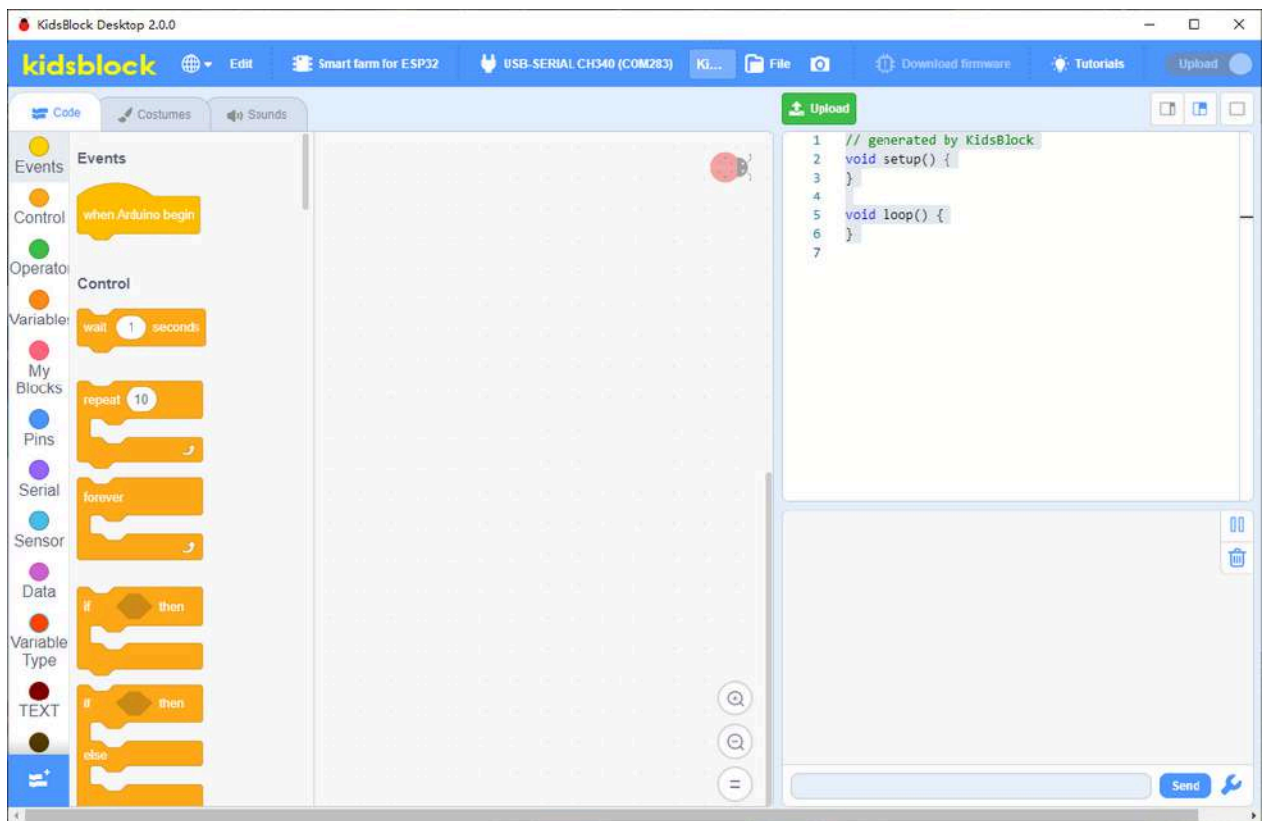
- După importul acestui kit, veți vedea următoarea interfață a portului Selectarea. Faceți clic pe **Conectare** la portul corect.




- Atingeți **Accesați editorul**.

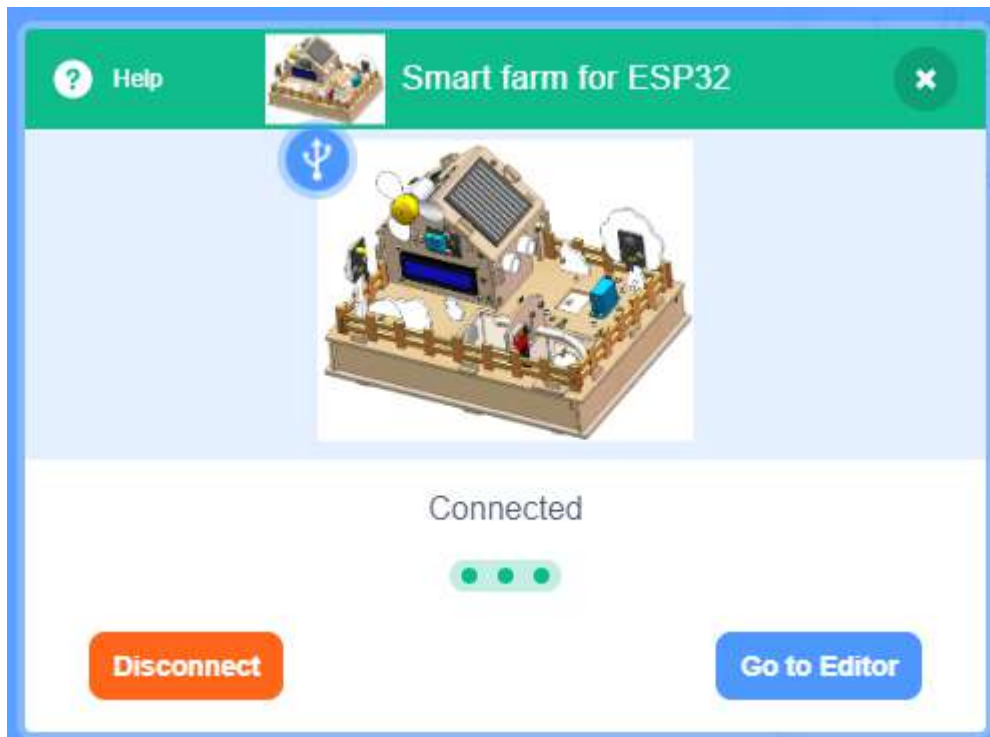


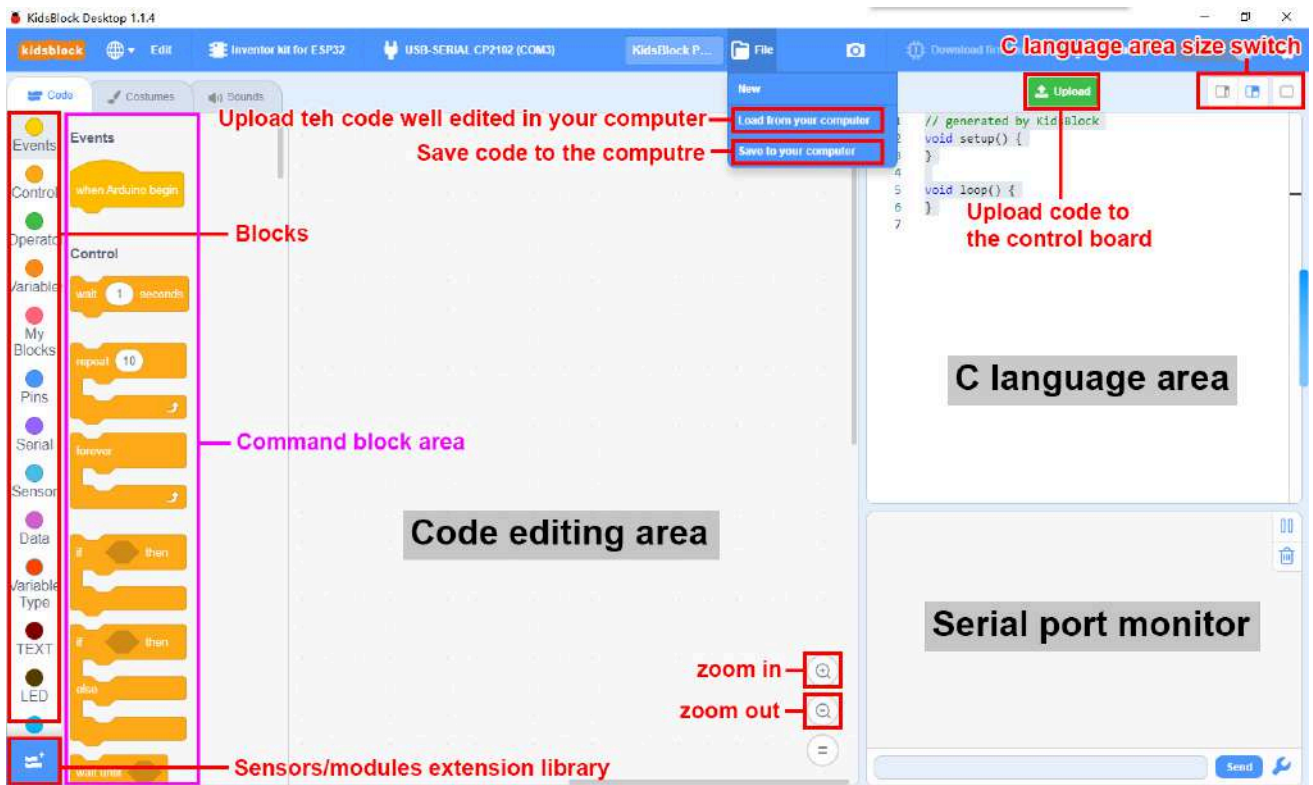
- Pagina principală:



Deconectați dispozitivul


- Dacă doriți să deconectați kitul și portul, faceți clic pe  **USB-SERIAL CH340 (COM283)**.
- Apoi atingeți **Deconectare** pentru a deconecta conexiunea curentă.

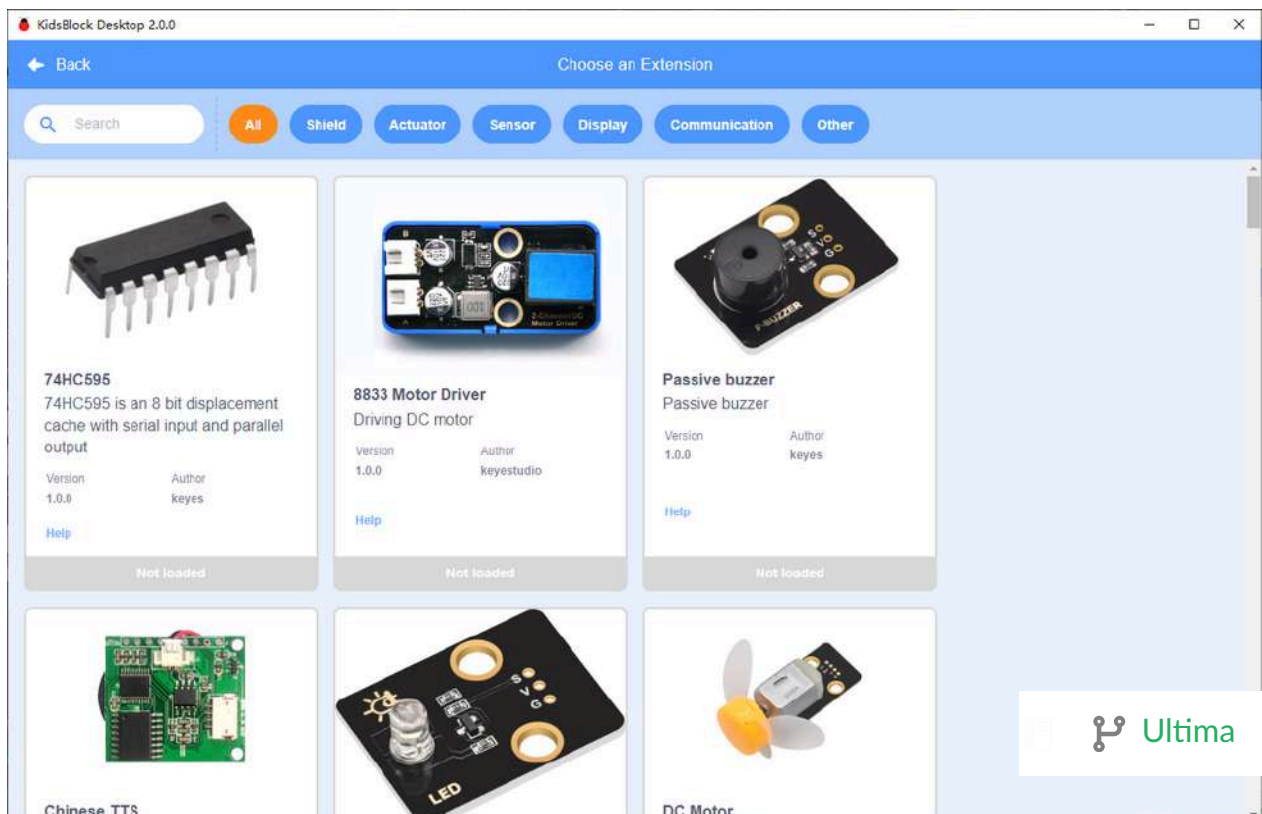





2.2.6 Extindeți un senzor/modul

NOTĂ: Această piesă poate fi sărită, deoarece toți senzorii necesari sunt integrați în kit fără a fi nevoie să vă extindeți. Dacă doriți să adoptați un , vă rugăm să consultați următoarele proceduri.

- Faceți clic  pentru a intra în biblioteca de extensii senzori/module.
- Alegeți o extensie.



- De exemplu, este necesar un modul de sonerie, vă rugăm să faceți clic pe pasiv Sonerie:




Passive buzzer
Passive buzzer

Version	Author
1.0.0	keyes

[Help](#)

Not loaded

- Când "Neîncărcat" devine "Încărcat", acest modul este importat cu succes.



Passive buzzer
Passive buzzer


Version	Author
1.0.0	keyes

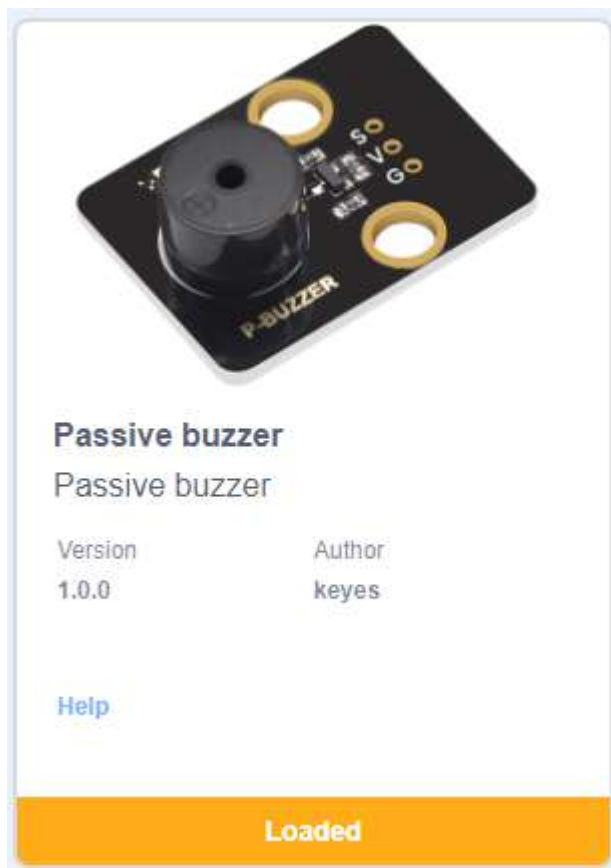
[Help](#)

Loaded

- Faceți clic [← Back](#) pentru a reveni la editor. Acum veți descoperi că un Blocarea soneriei pasive apare în Cod.




- Dacă doriți să eliminați "Buzzerul pasiv", trebuie doar să faceți clic  pe Introduceți Bibliotecă și atingeți.

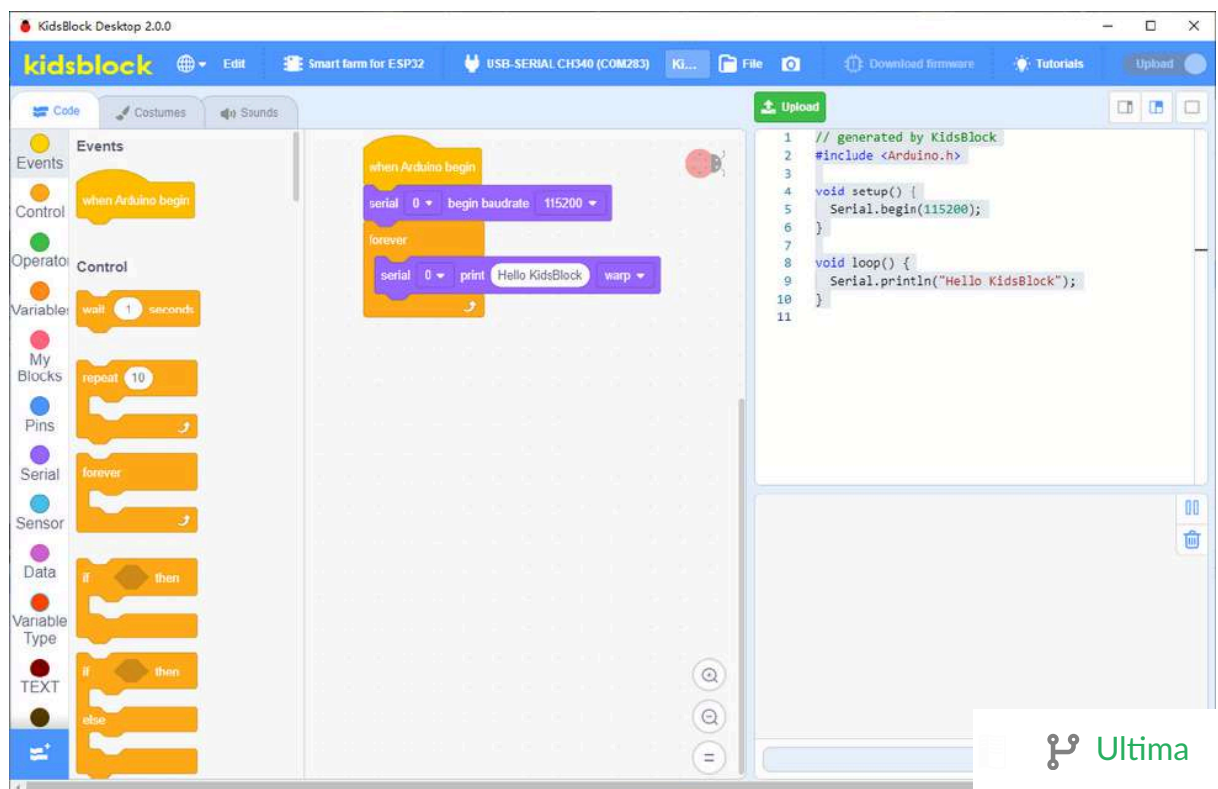


- Când "Încărcat" trece la "Neîncărcat", acest modul este reușit Șterse.



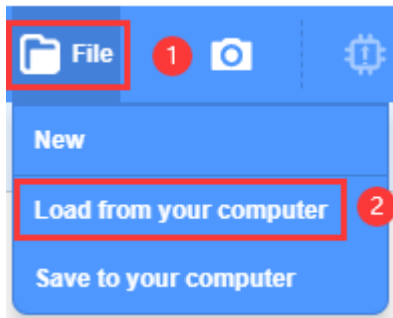
2.2.7 Importați fișierul

- Metoda 1
 - Dacă software-ul nu este în funcțiune, faceți clic direct pe fișierul SB3 pentru a-l deschide. De exemplu, faceți clic  `hello keystudio.sb3` pentru a-l deschide. Nu uitați să Selectați un dispozitiv.

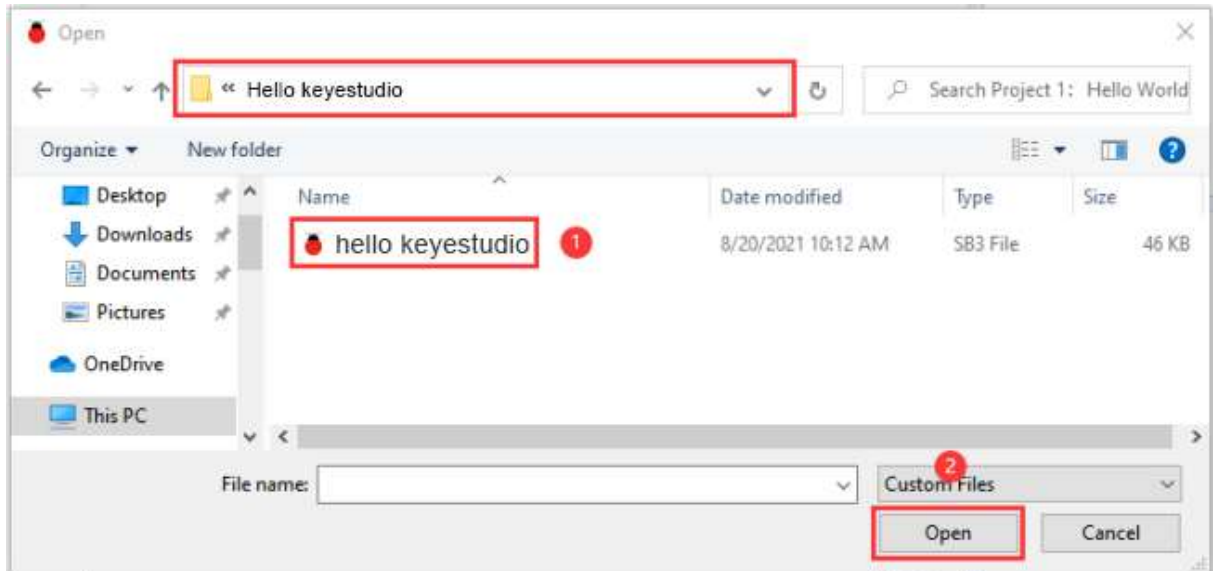


- Metoda 2

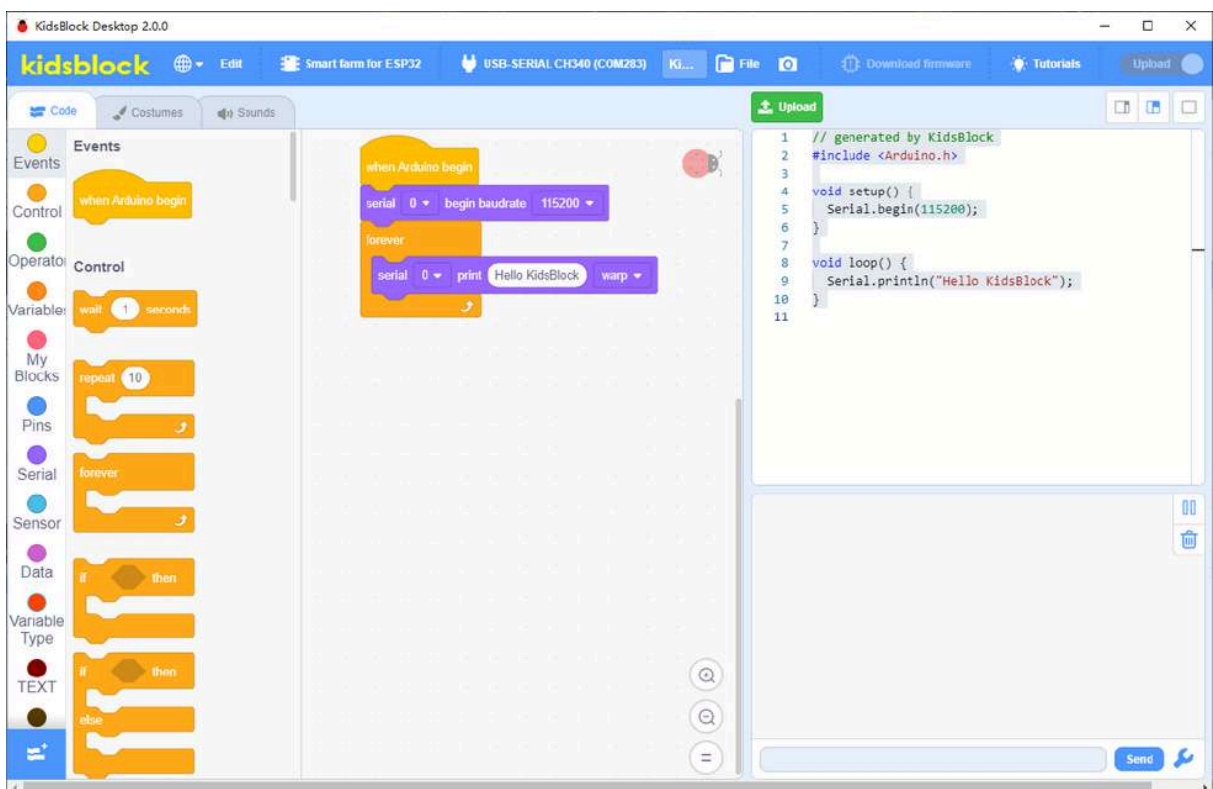
- Deschideți Kidsblock. Faceți clic pe "fișier" pentru a alege "Încărcați din computer".



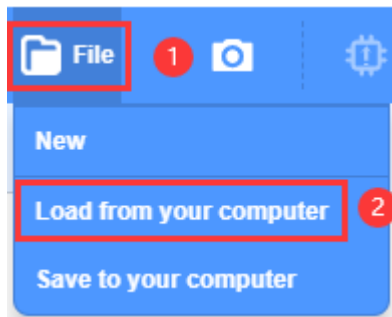
- Selectați un fișier SB3 (cum ar fi 🐞 hello keystudio.sb3).




- Importați cu succes!

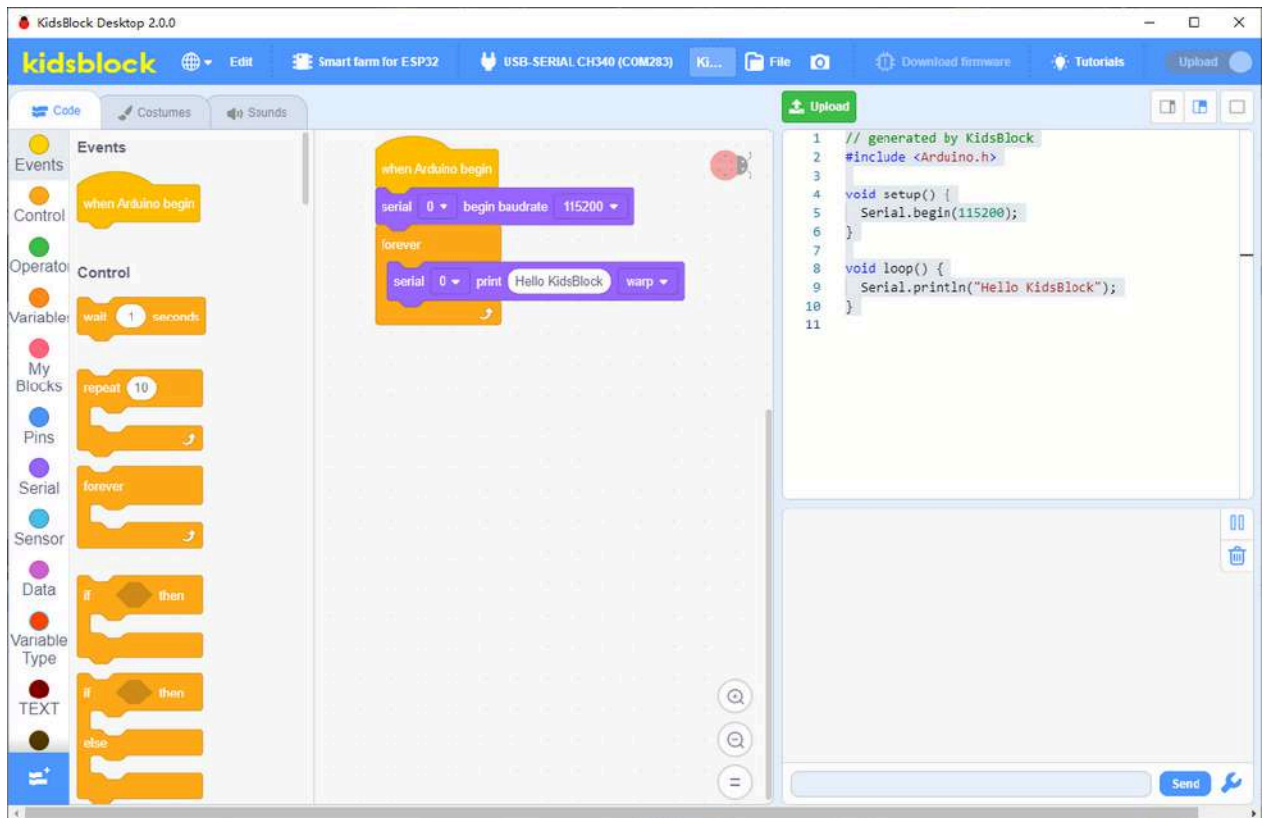


2.2.8 Încărcați codul și setați rata de transmisie



- Încărcați fișierul în Kidsblock.

- Conectați placa de dezvoltare la computer (dacă nu puteți vedea un port, vă rugăm să instalați mai întâi un driver) și selectați portul corect și faceți clic pe  .



- Așteptați încărcarea.






Setarea ratei de transmisie

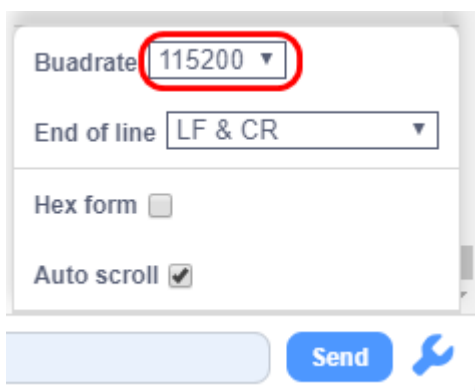
- Dacă nu există o casetă de imprimare, vă rugăm să faceți clic pe una dintre



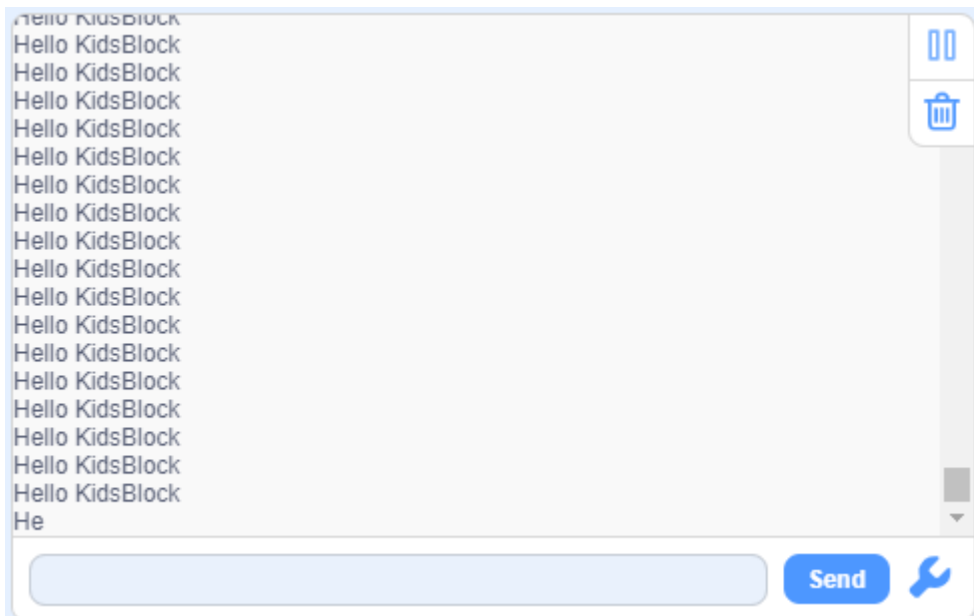
pentru a ajusta dimensiunea cutiei.

- Cutie de imprimare mică 
- Cutie mare de imprimare 
- Fără cutie de imprimare 

- Faceți clic  pentru a seta rata de transmisie corespunzătoare.



- "Hello KidsBlock" începe să fie tipărit în cutie după ce setare.



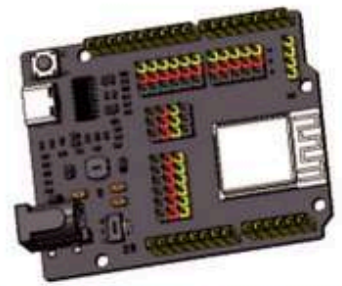
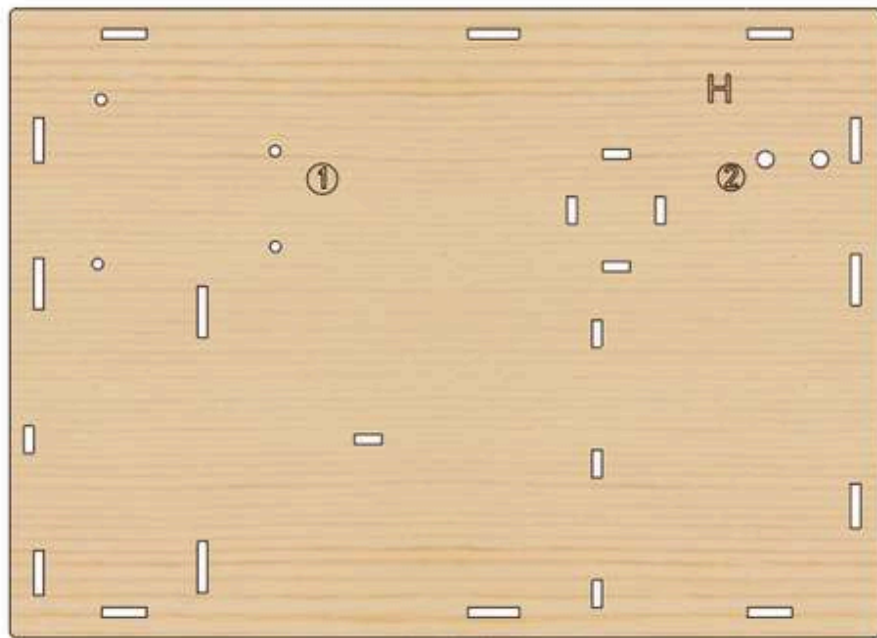
3. Asamblare

În timpul asamblării, unele coduri trebuie arse, așa că vă rugăm Instalați mai întâi software-ul.

Întregul ansamblu poate fi împărțit în două părți: asamblare piese și cablaje.

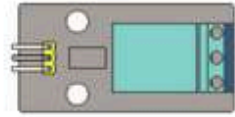
Pasul 1 Instalați placa ESP32 și modulul releu

1.1 Componente necesare



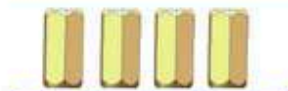
Keyestudio ESP32 PLUS Development Board

×1



Relay Module

×1



M3*10MM Dual-pass Copper Pillar

×4



M3*6MM Round Head Screw

×8



M4*8MM Round Head Screw

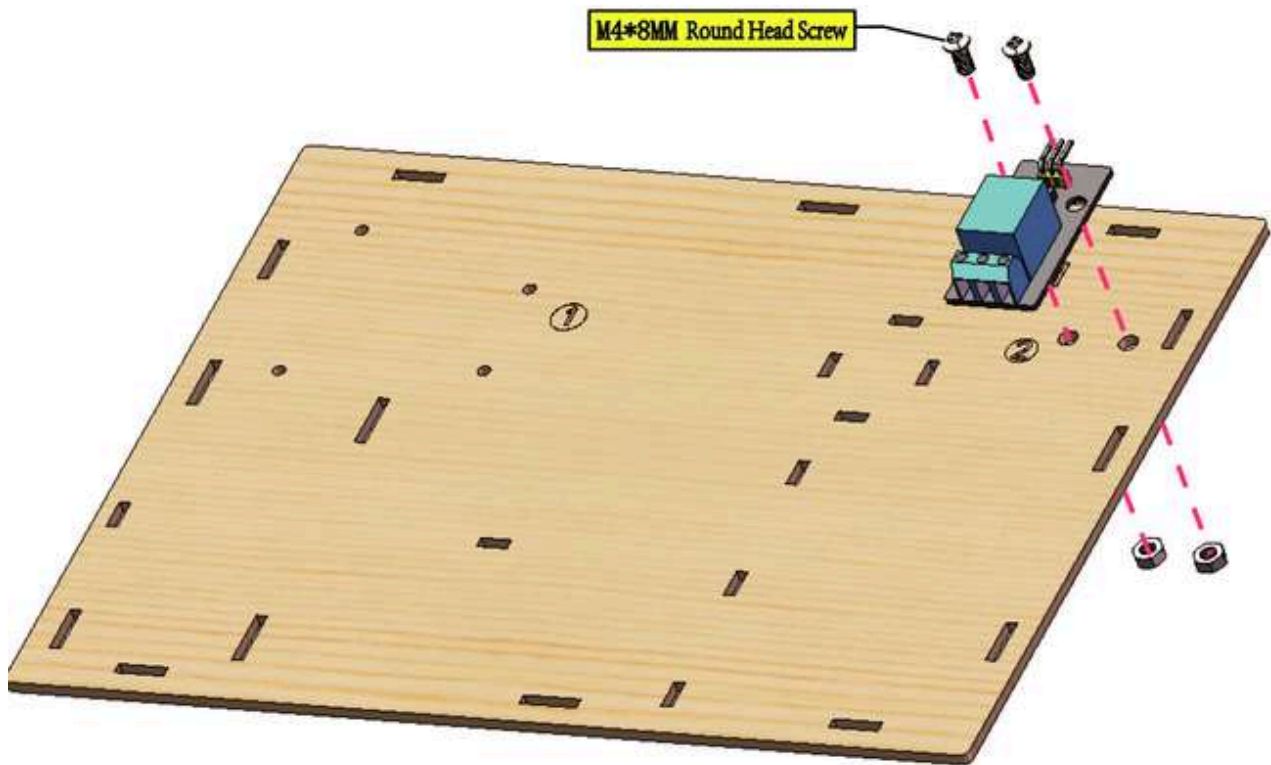
×2



M4 Nut

×2

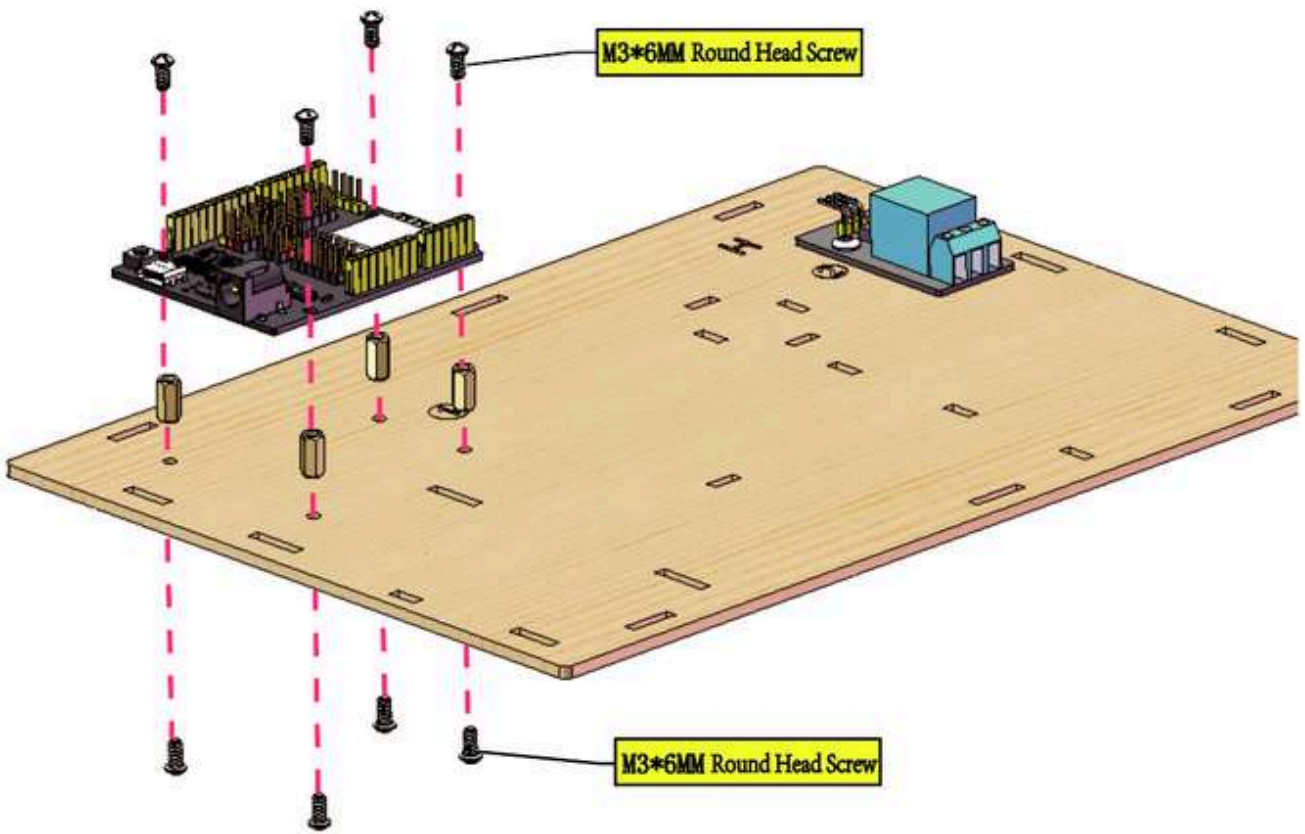
1.2



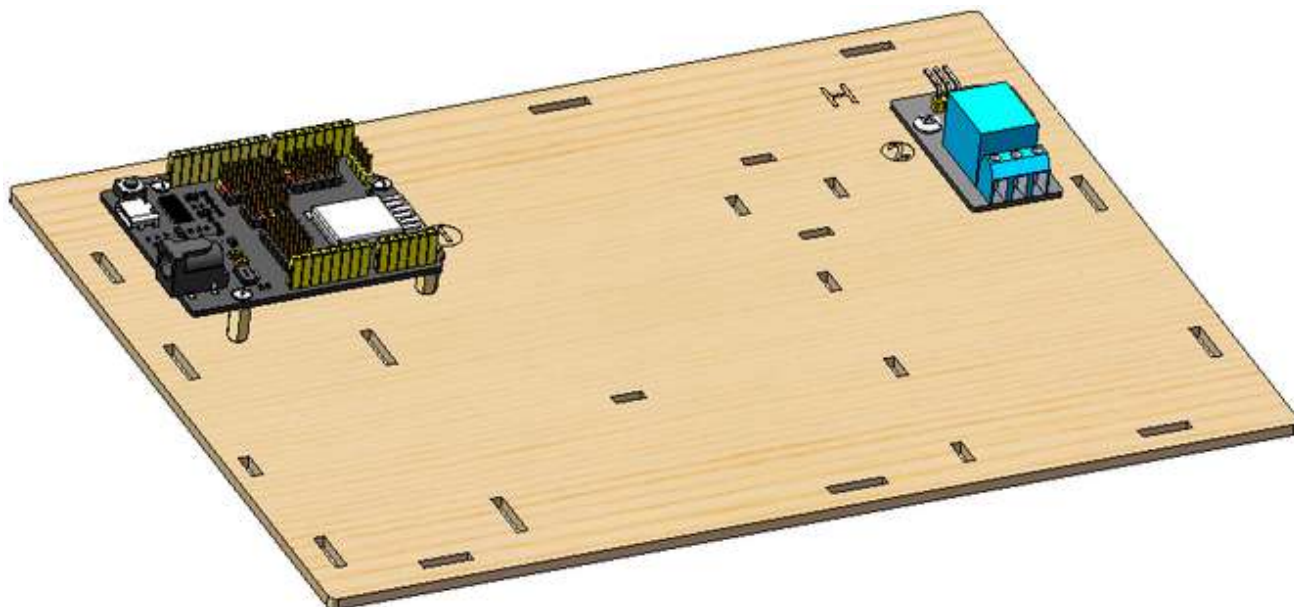
1.3



1.4



1.5



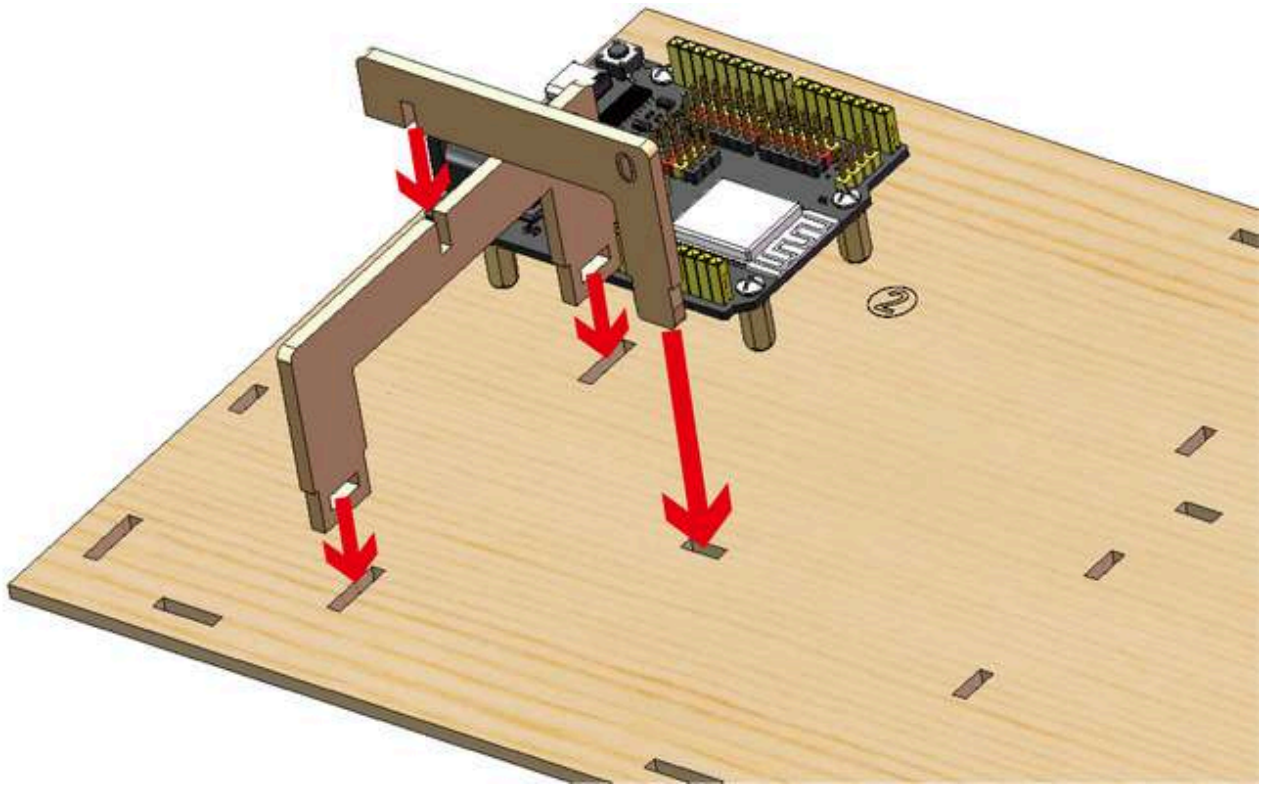
Pasul 2 Instalați cadrul de fixare pentru carcasa bateriei și instalați cabina de alimentare, conectați placa ESP32 și modulul releu

2.1 Componente necesare

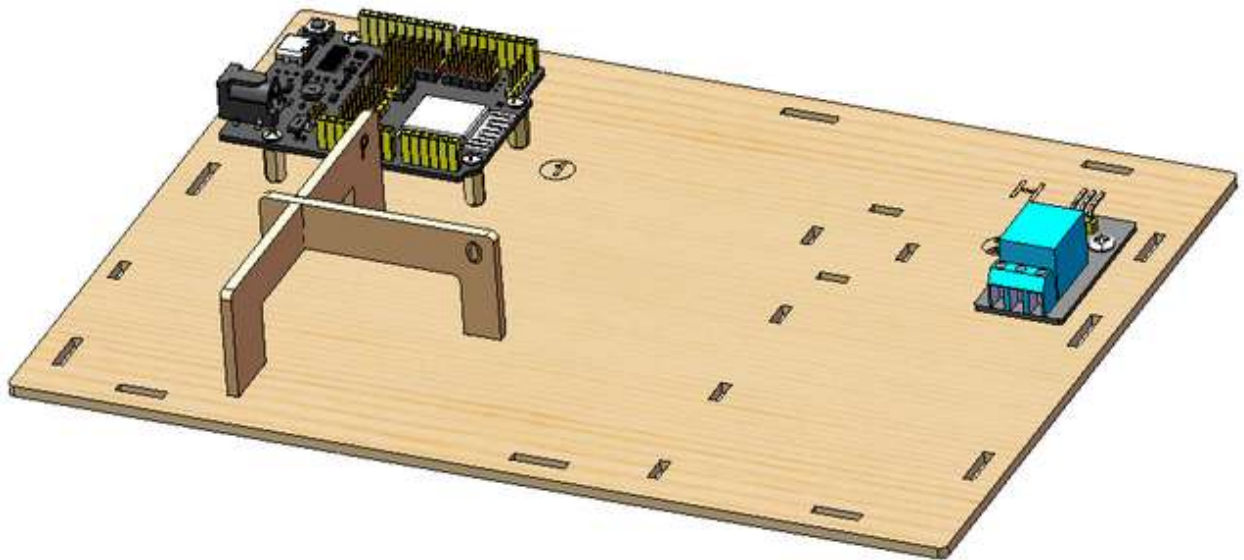


2.2

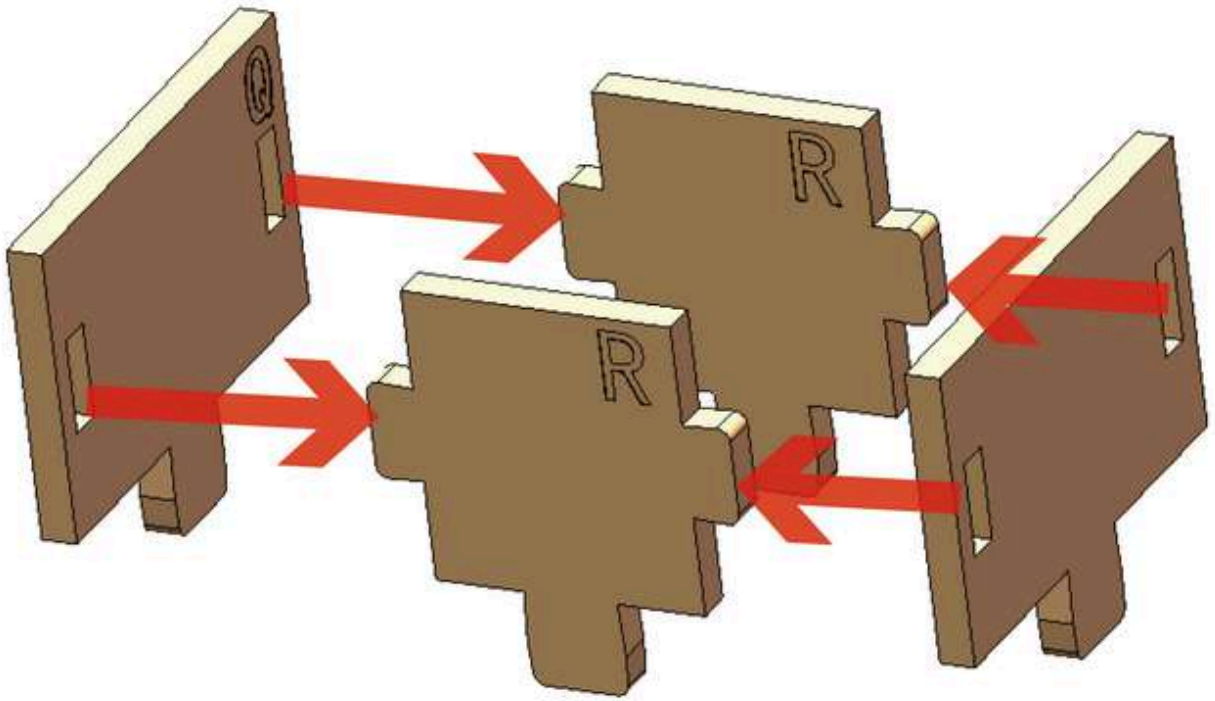
Asamblați placa de lemn X și O pe placa inferioară



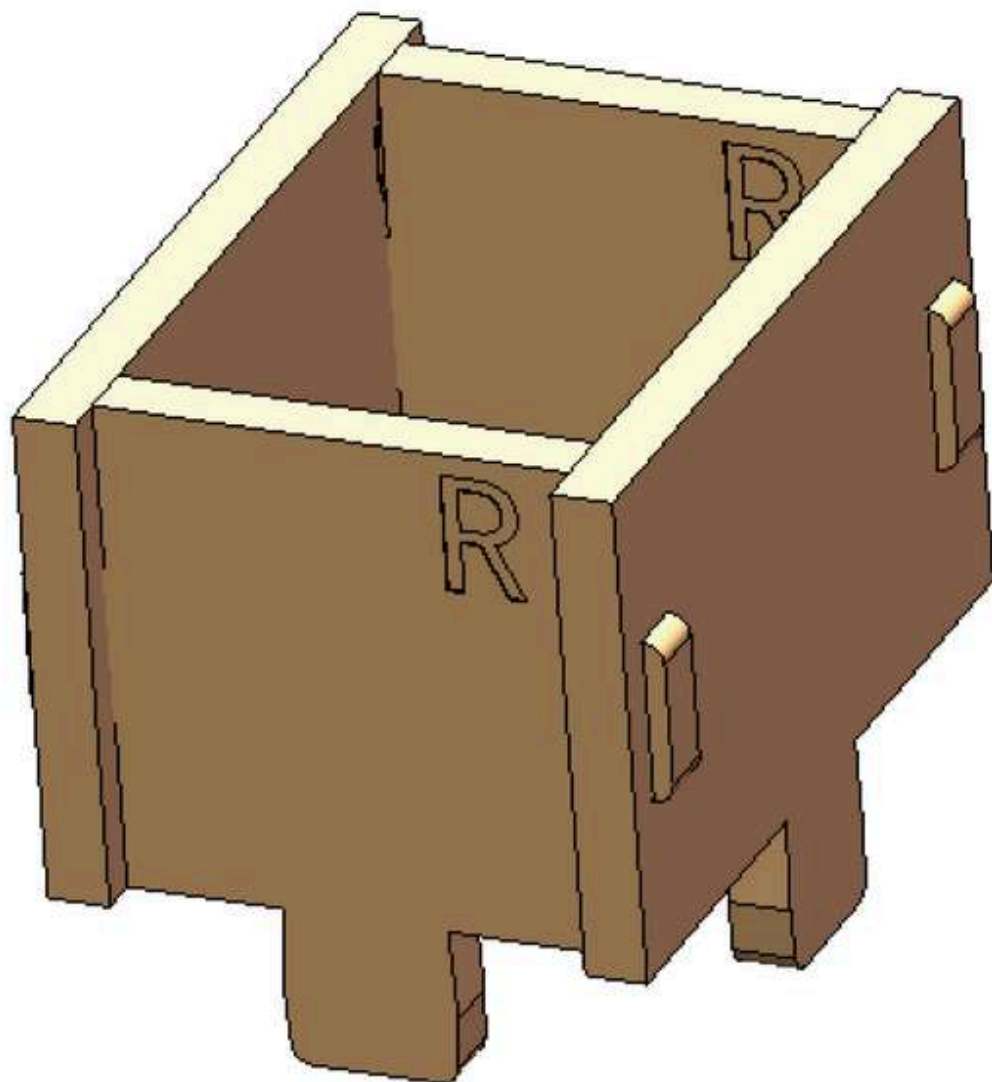
2.3



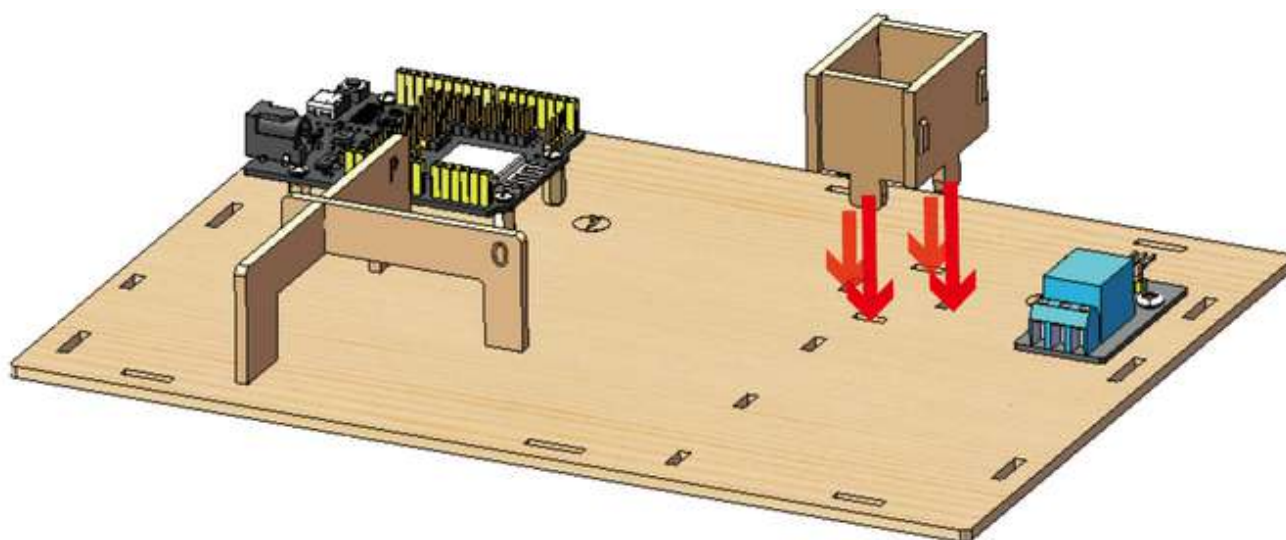
2.4



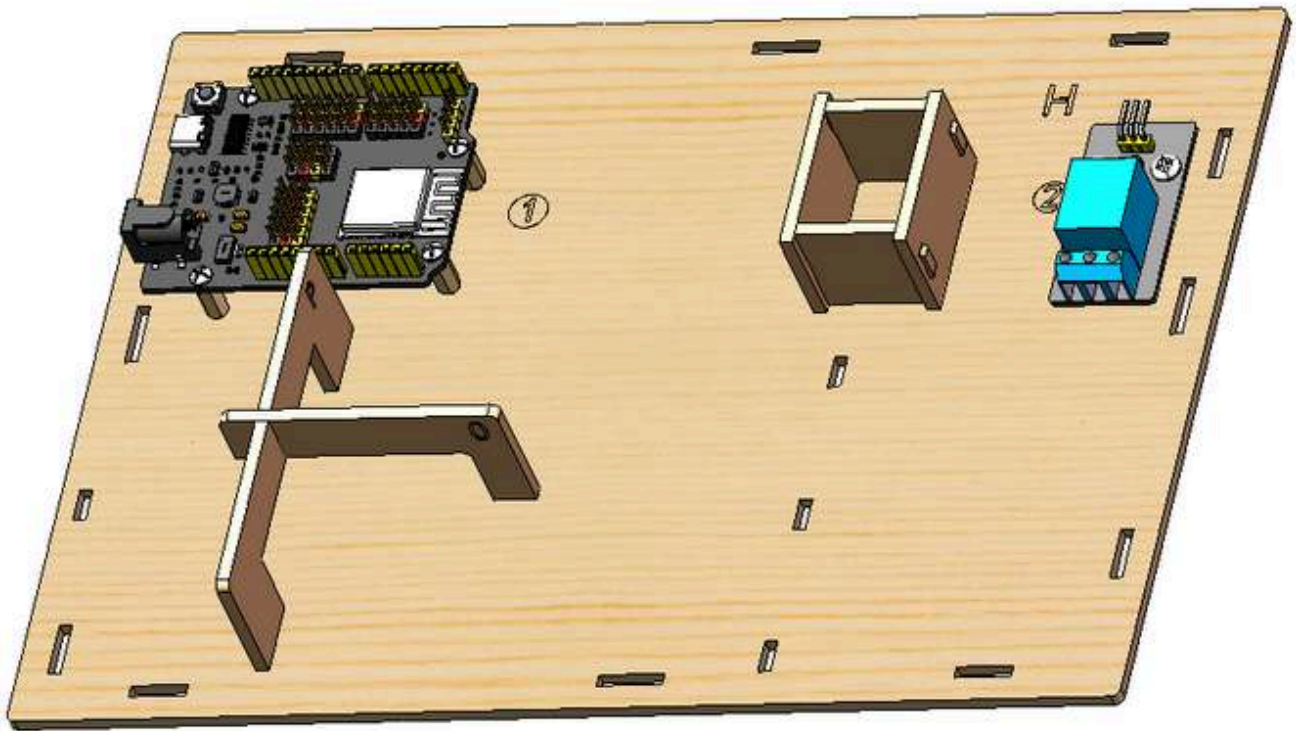
2.5



2.6



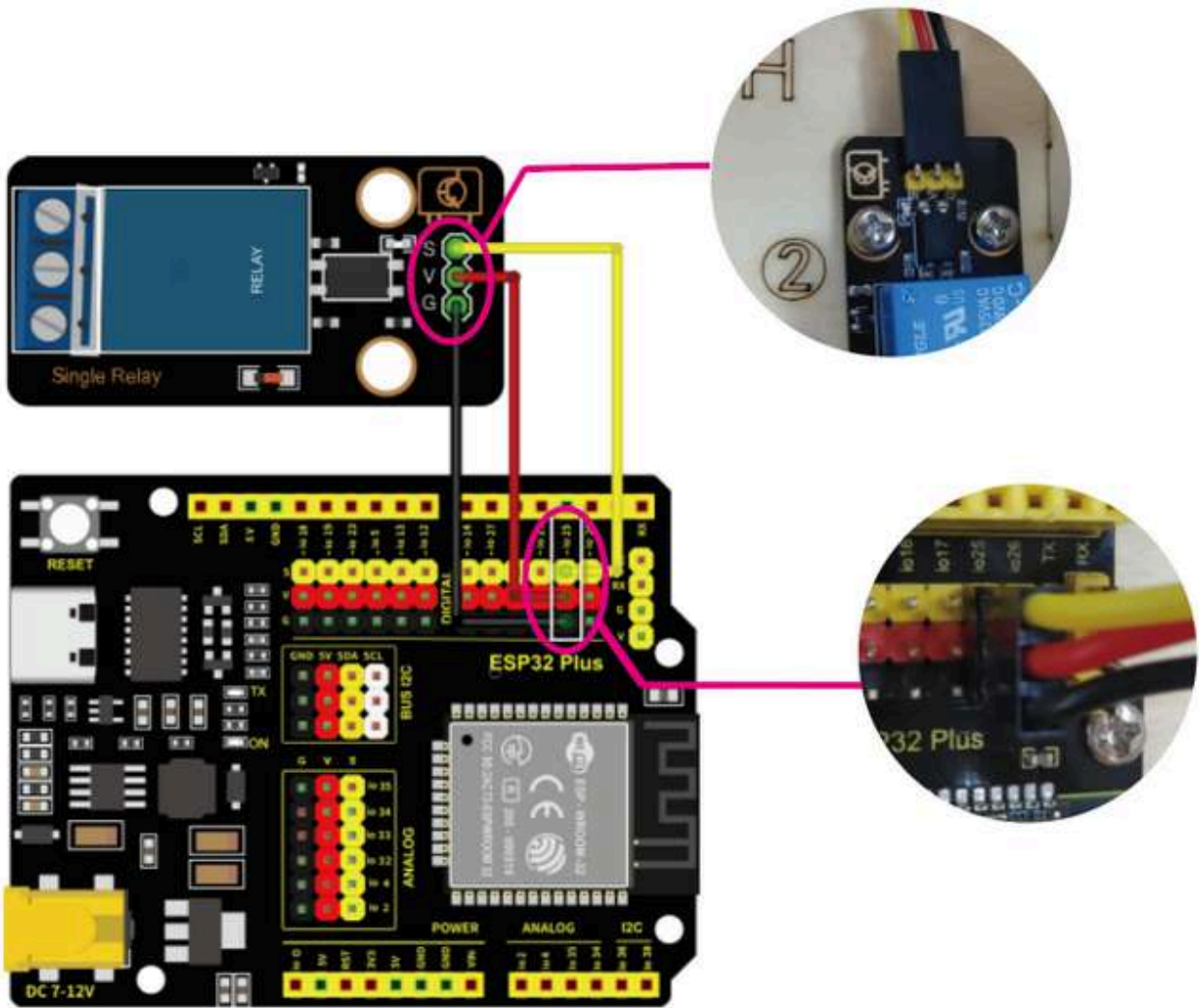
2.7



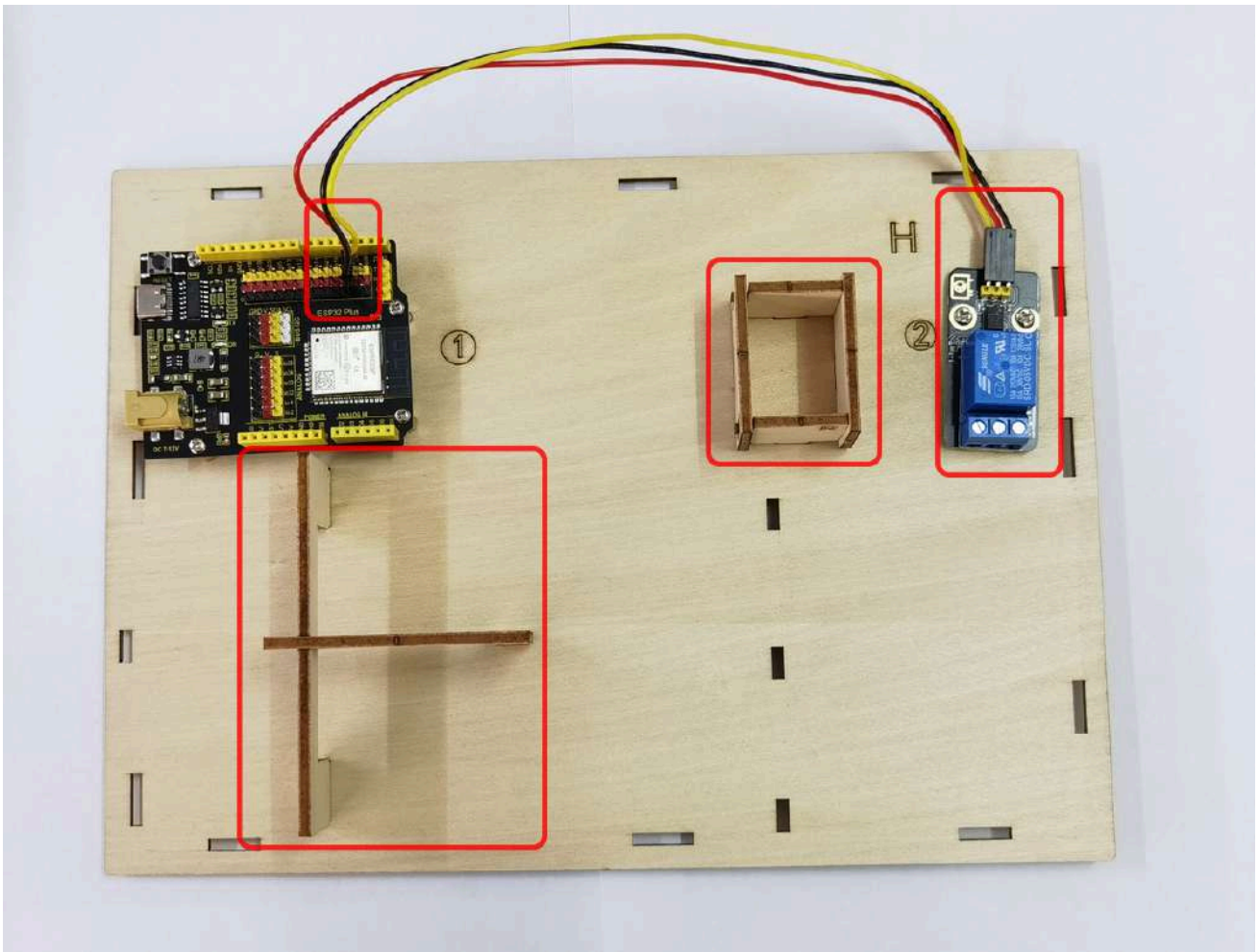
2.8 Conectați placa ESP32 și modulul de releu

Modul	Sârmă	Pin
Modul releu	3PIN 20cm	IO25

Pin modul	Culoarea sârmei	Pin placă ESP32
V	ROȘU	V
G	NEGRU	G
S	GALBEN	io25

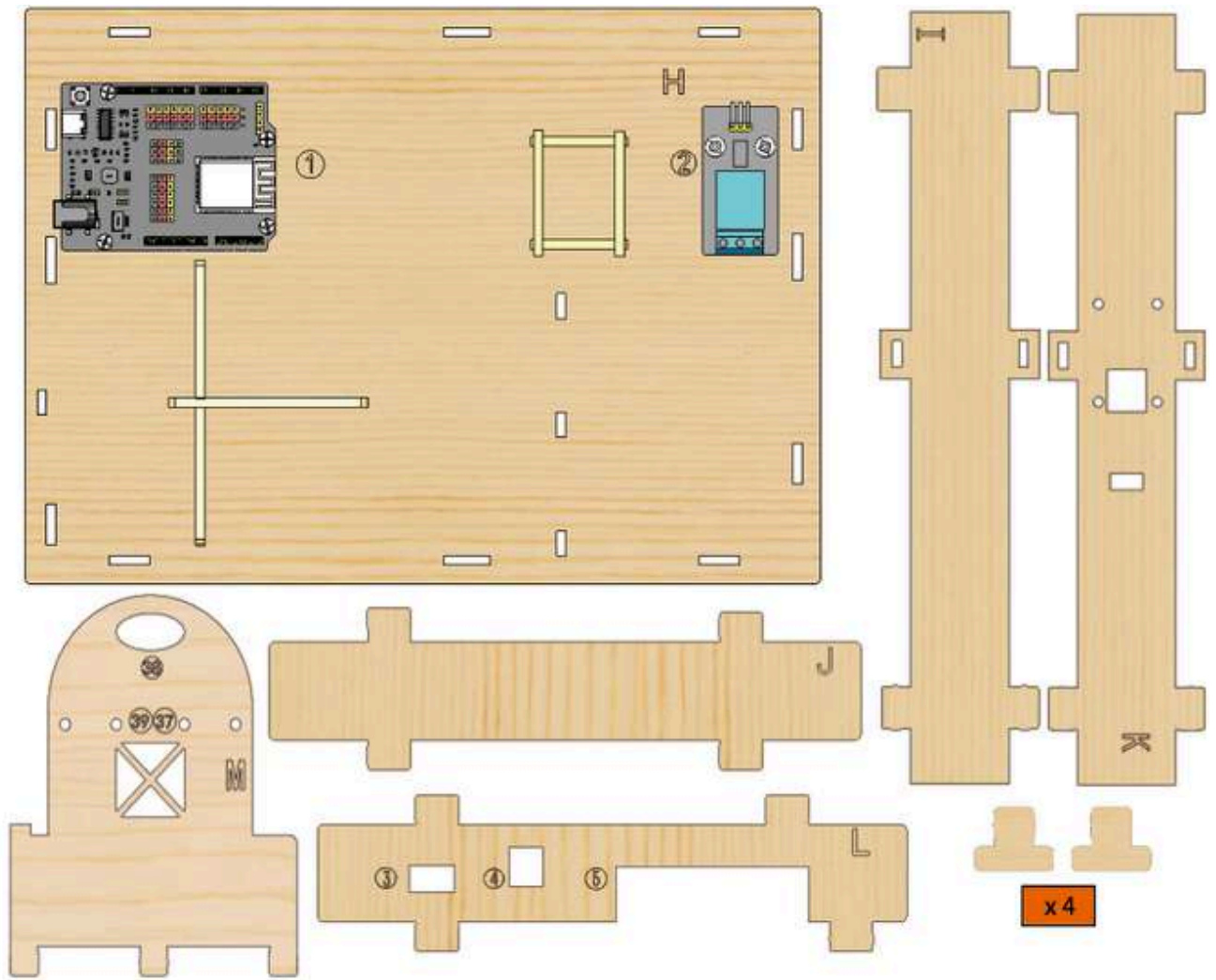


2.9

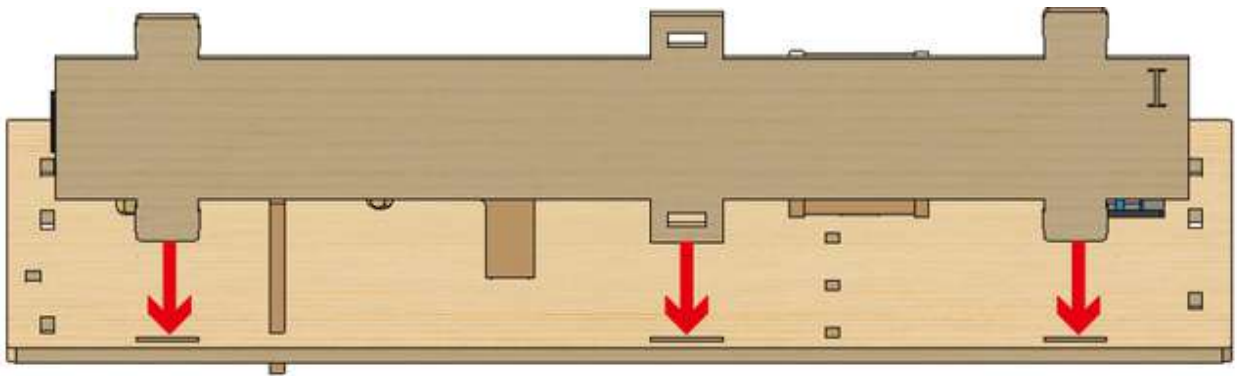


Pasul 3 Instalați substructura casei

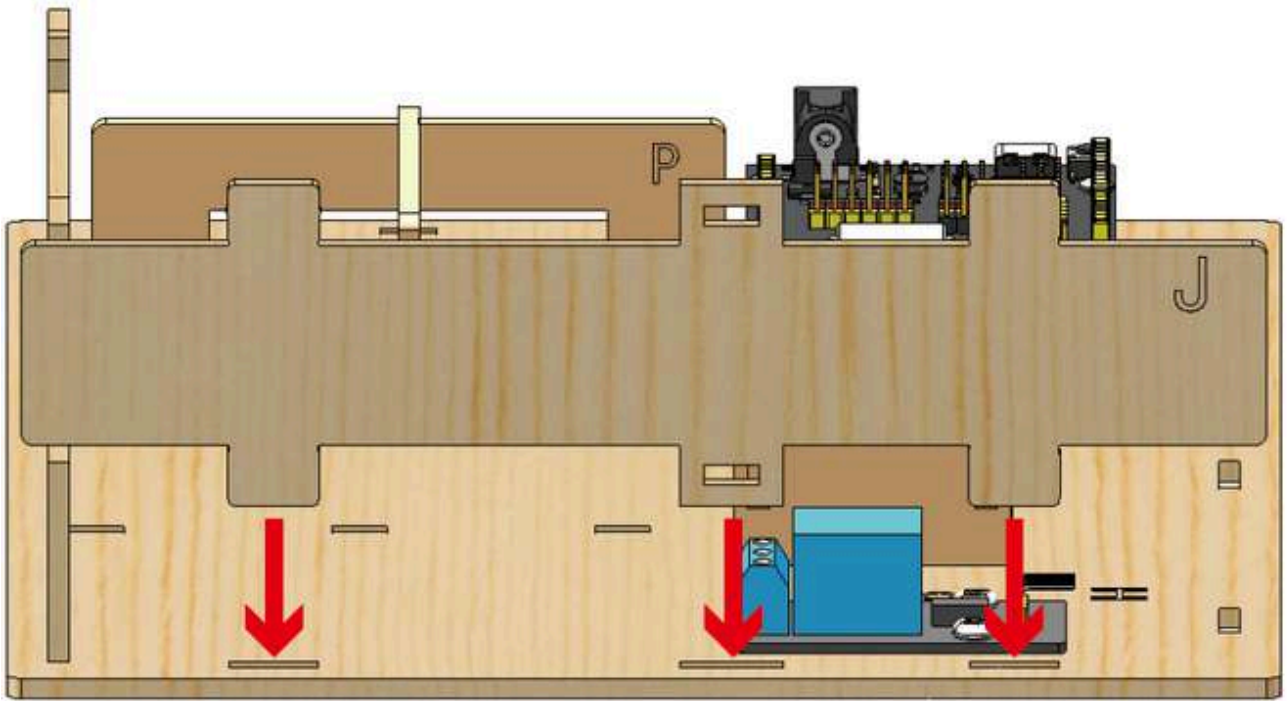
3.1 Componente necesare



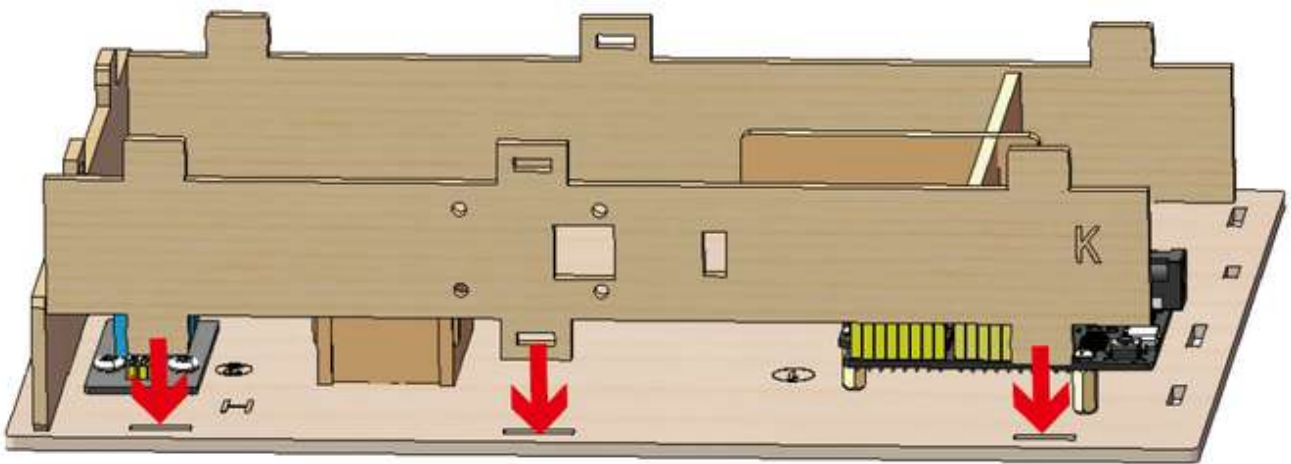
3.2



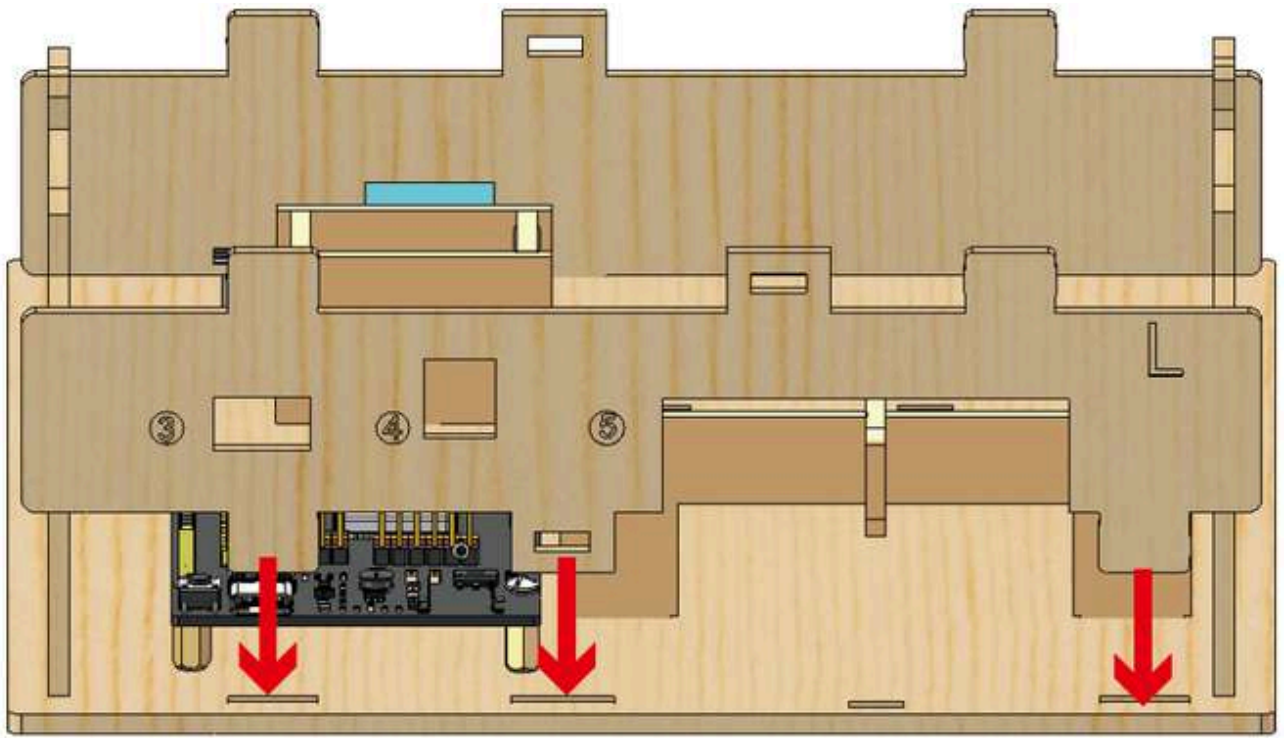
3.3



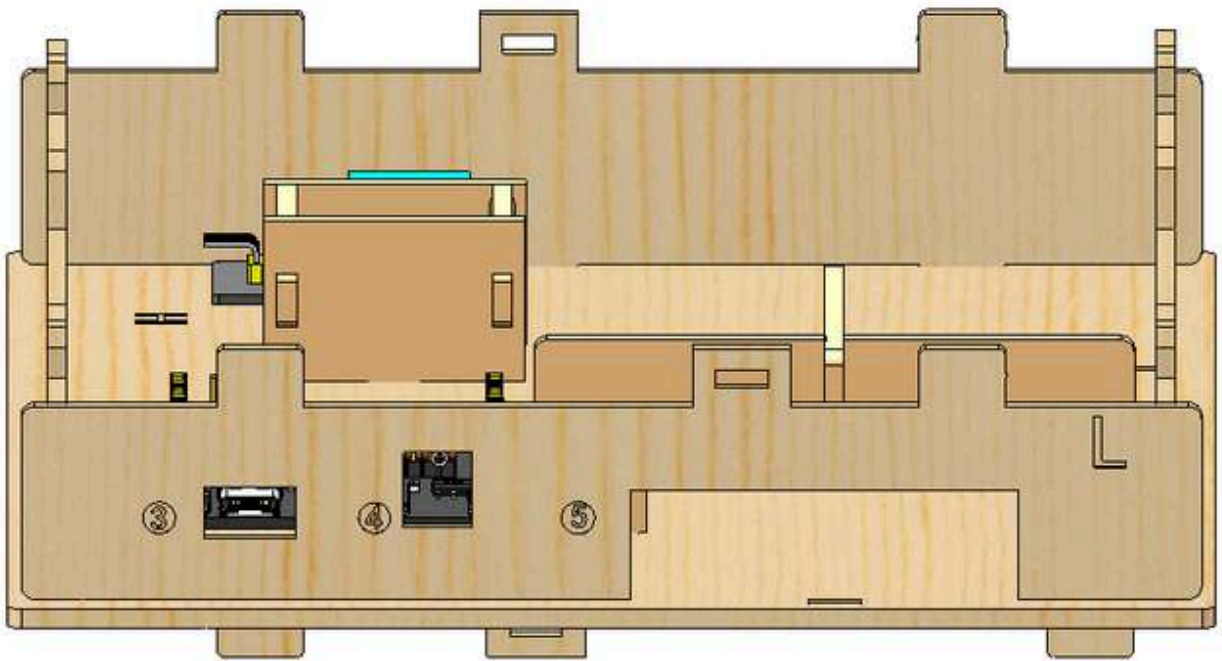
3.4



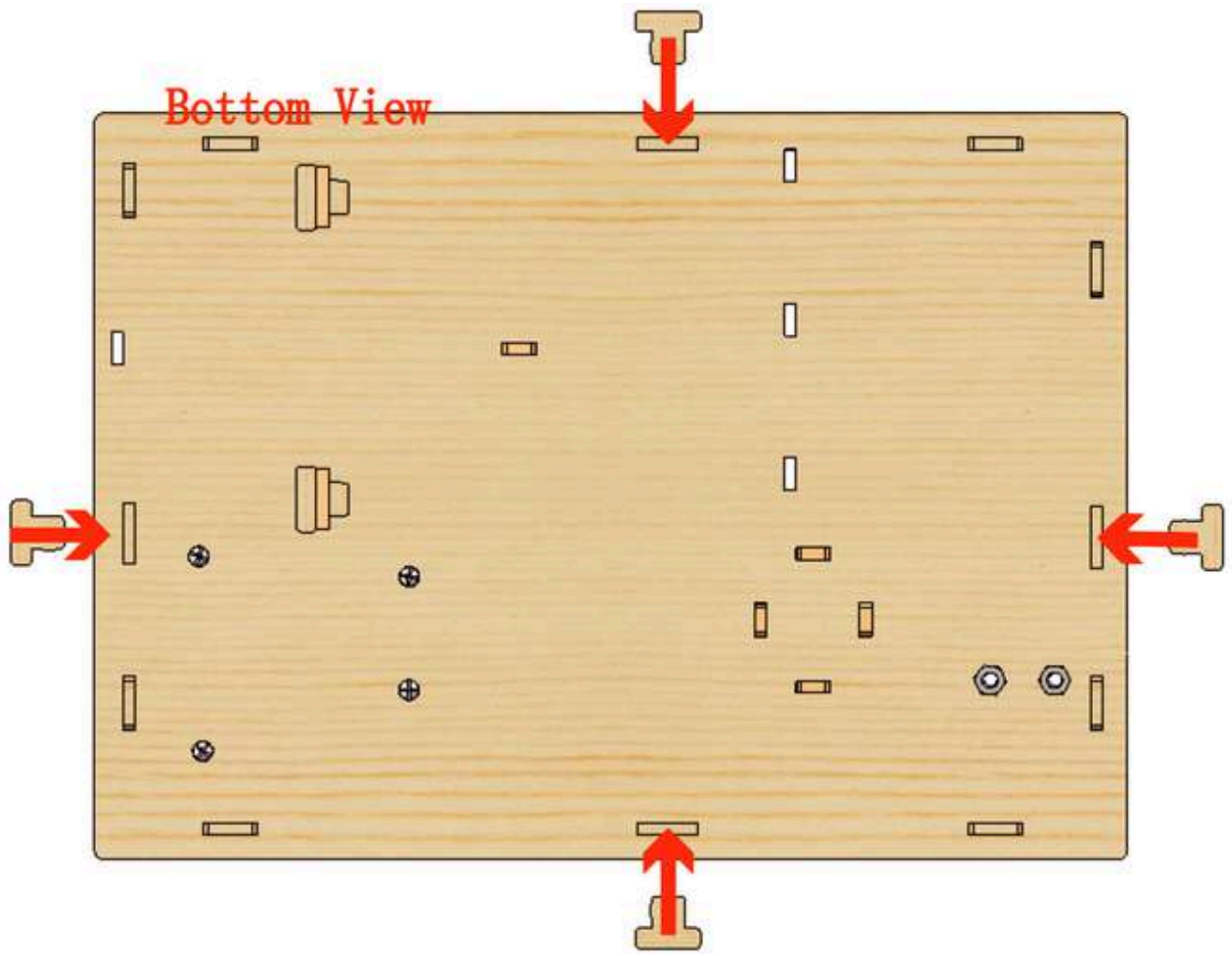
3.5



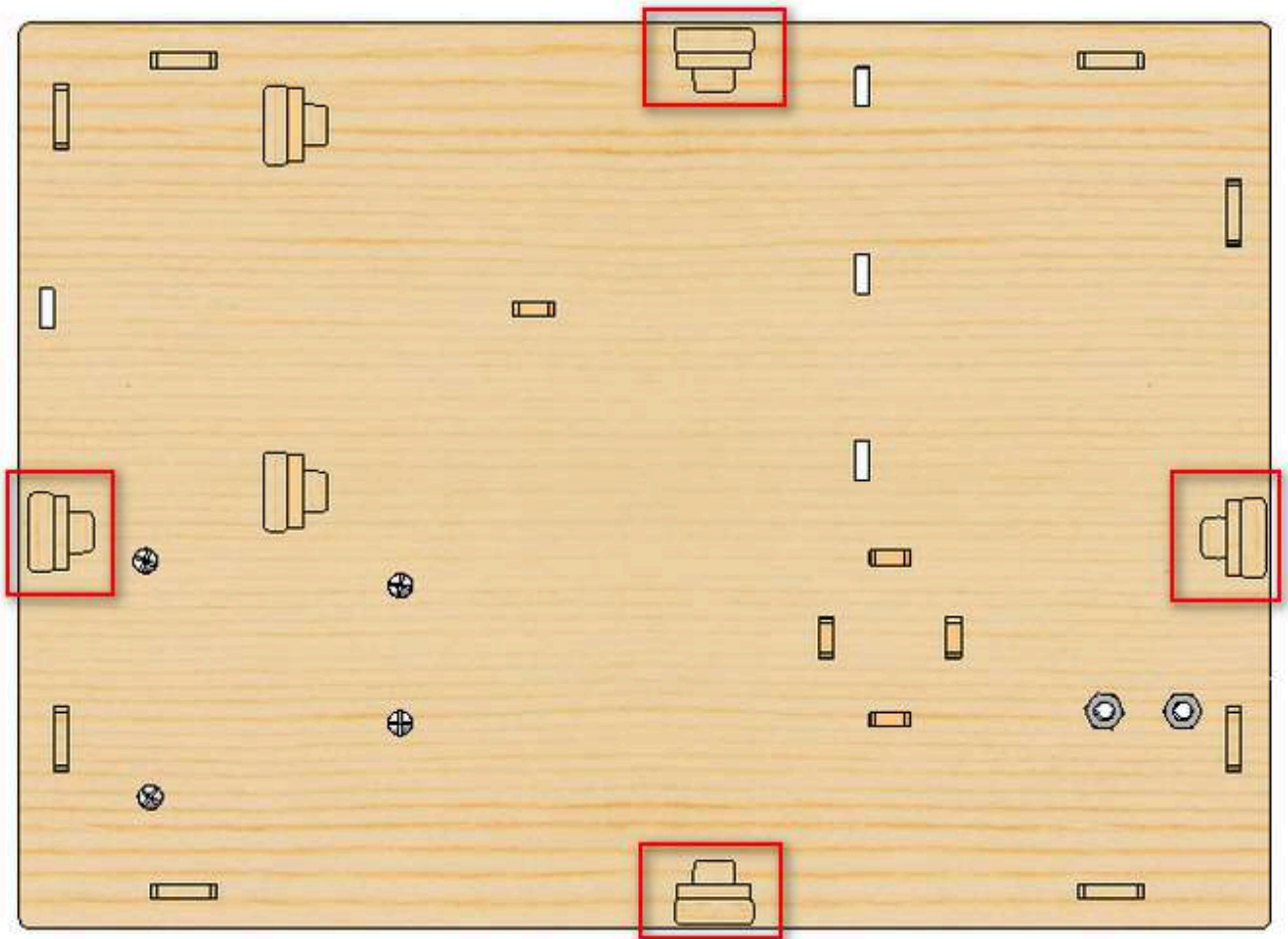
3.6



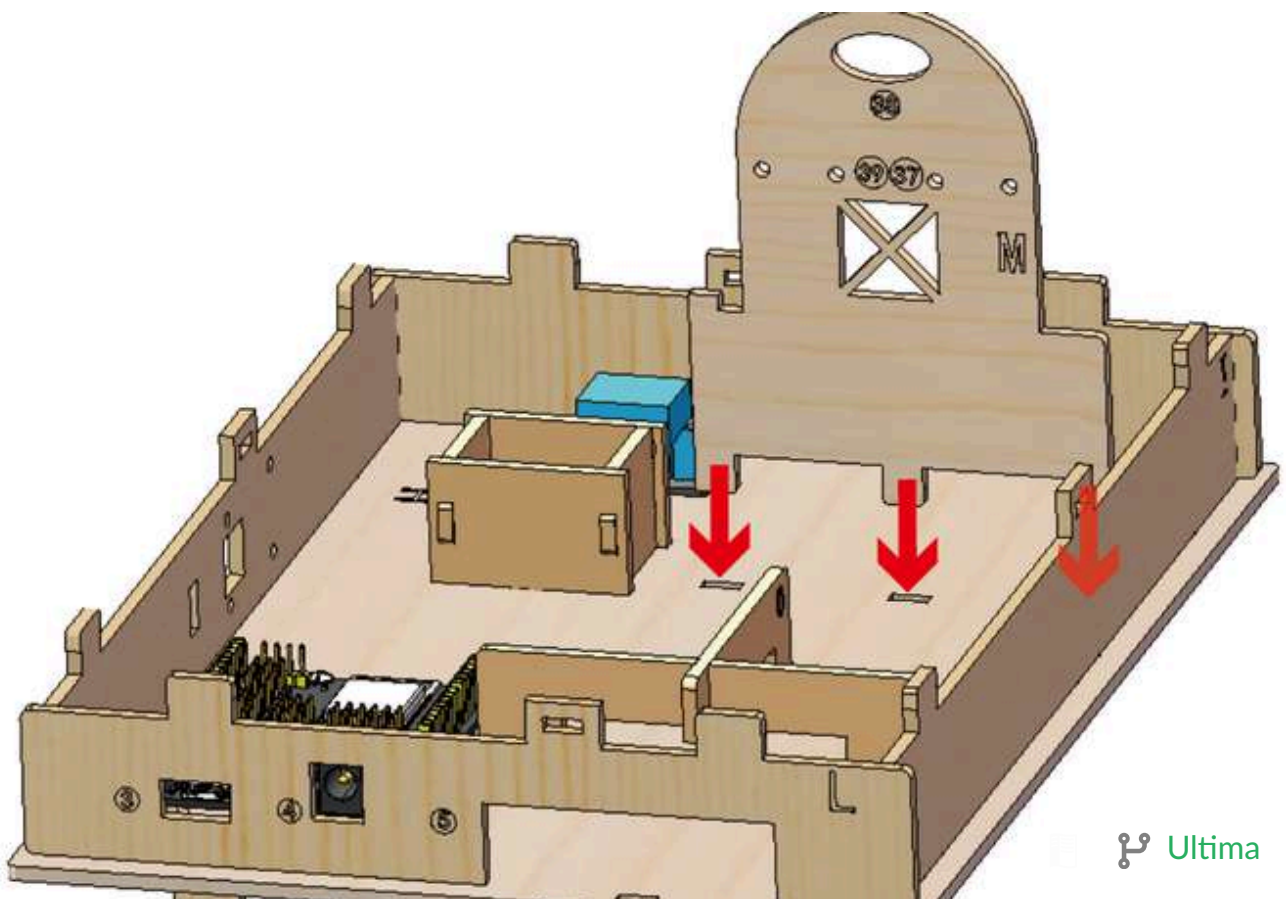
3.7



3.8

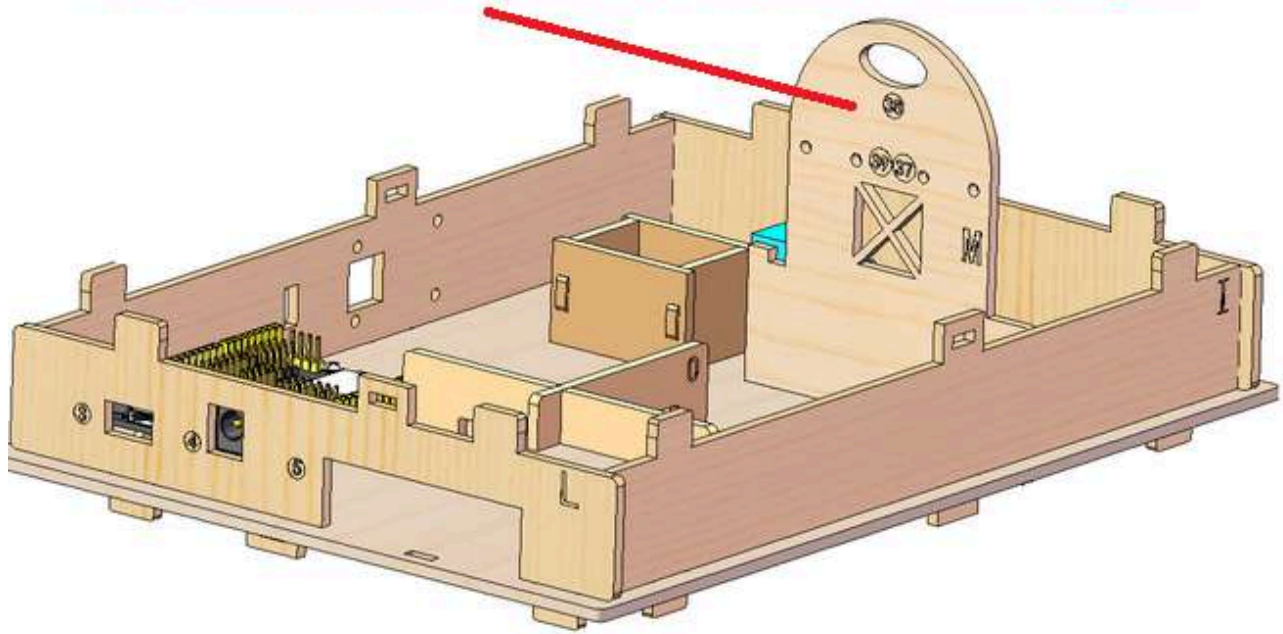


3.9

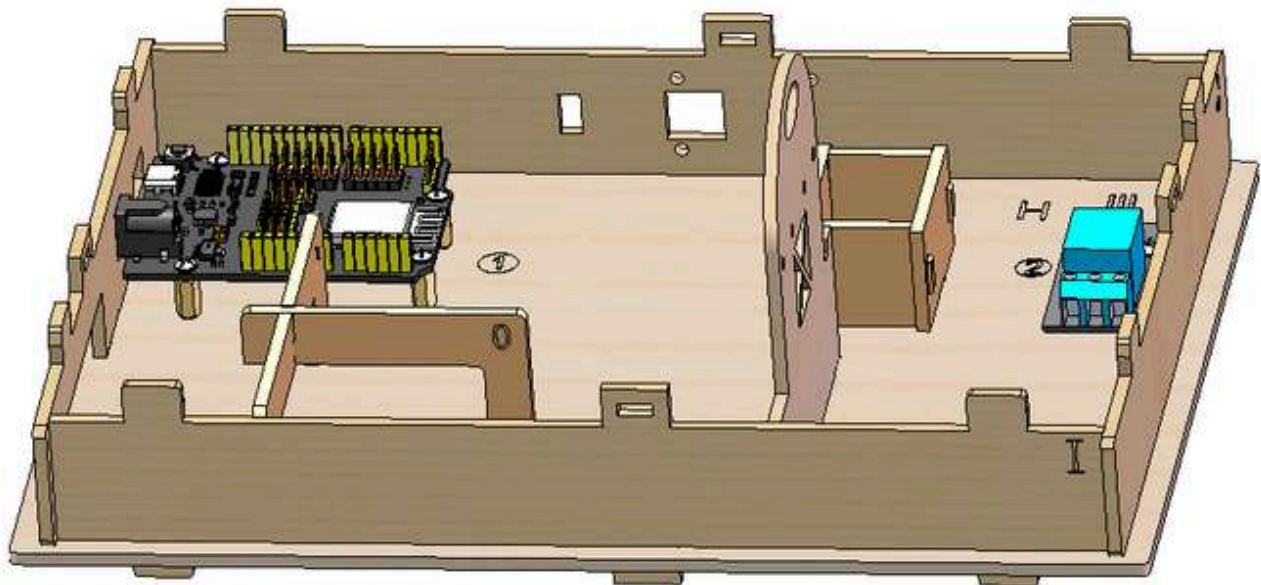


3.10

Note the installation direction of the board, with M facing us.

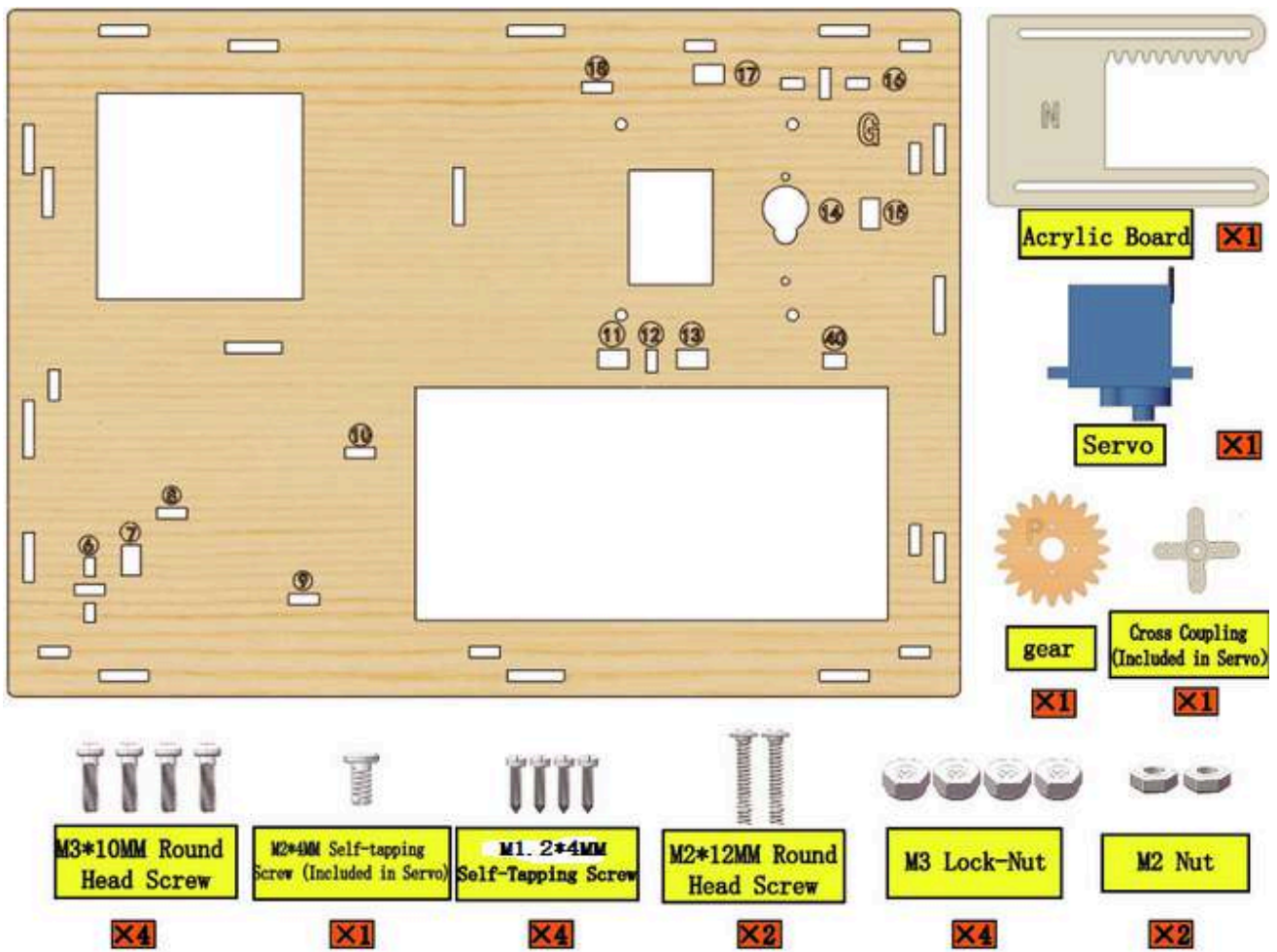


3.11



Pasul 4 Instalați ușa cabinei de hrănire

4.1 Componente necesare



Foia acrilică este ambalată separat și este recomandat să rupeți folia de protecție pentru a reduce frecarea atunci când se mișcă ca ușă.

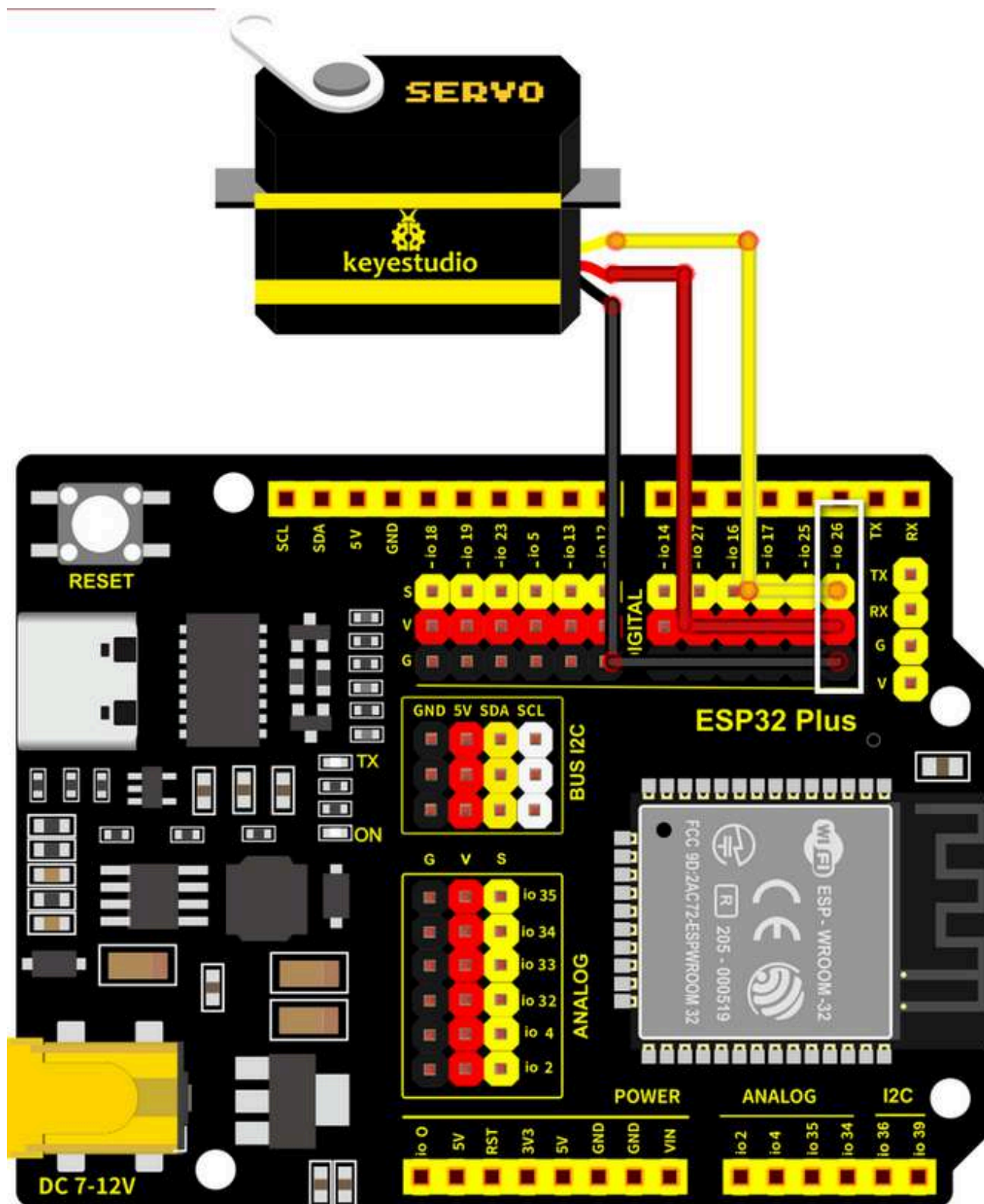
4.2 Setați servo la 180°

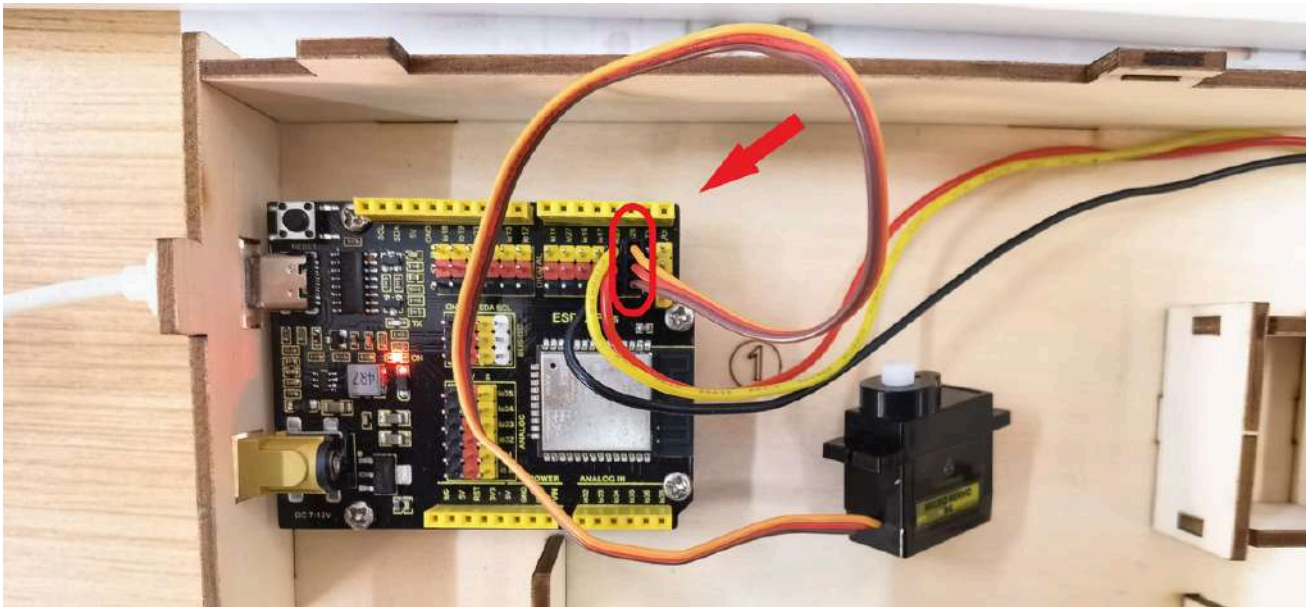
Vă rugăm să rețineți că acest pas este foarte important. Înainte de instalare, asigurați-vă că setați servo la 180 °.

Înainte de a monta servo, în primul rând vă rugăm să setați unghiul la 180 ° KidsBlock (Programare grafică).

Vă rugăm să consultați pașii de instalare a software-ului pentru detalii.

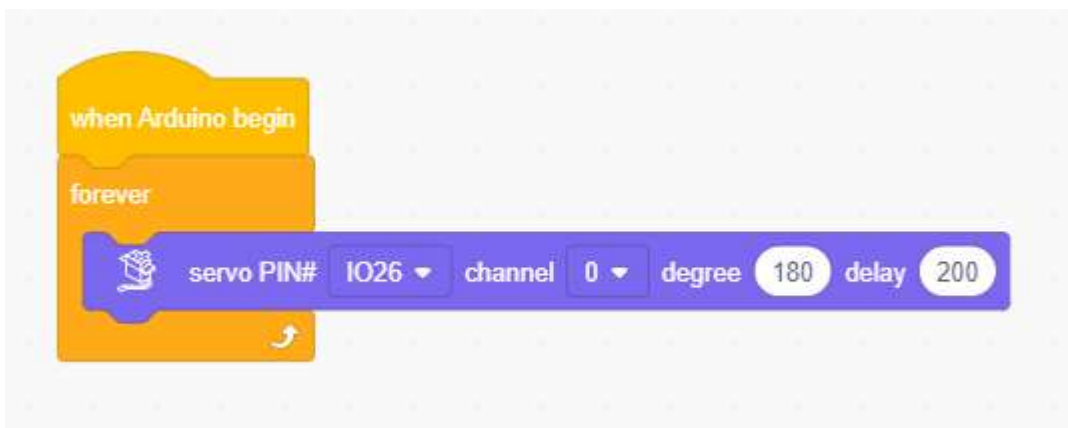
Conectați Servo la placa de dezvoltare ESP32 la pinul IO26.





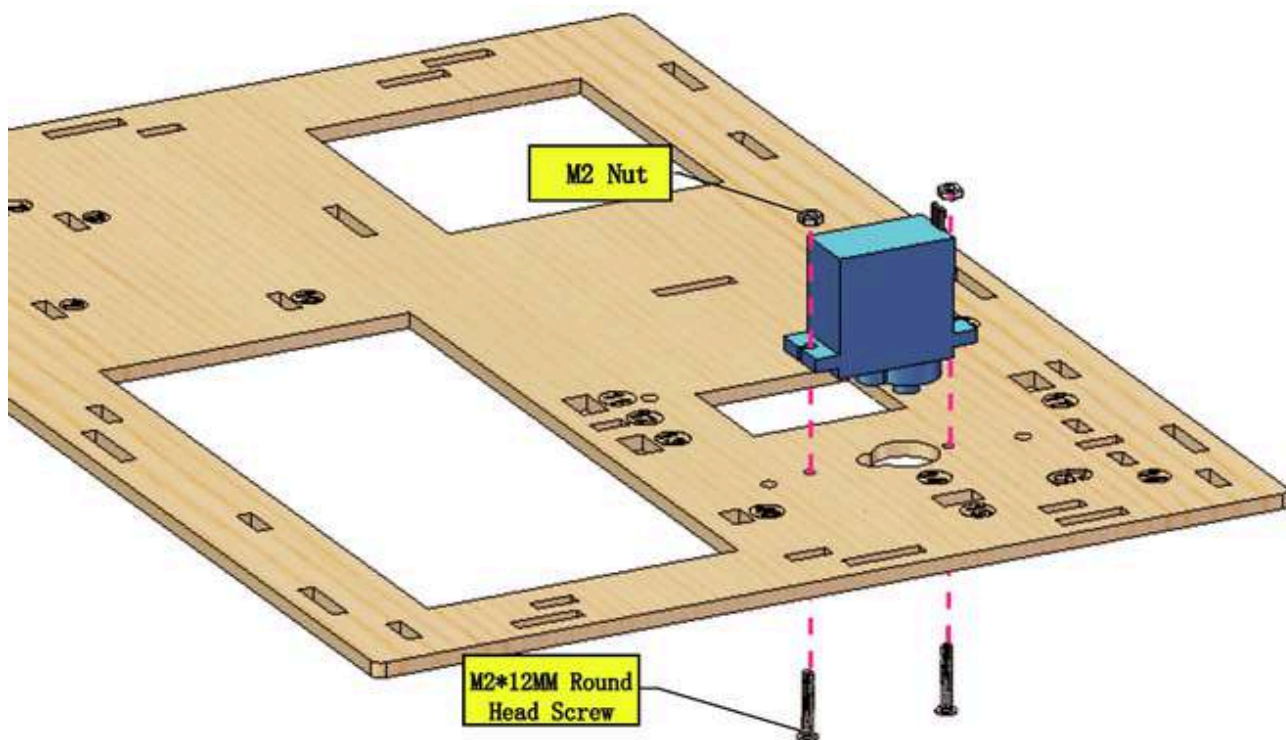
- Deschideți KidsBlock. Selectați placa ESP32 și portul serial corect, și inscripționați următorul cod pe tablă.

Name	Date modified
hello Keystudio	4/25/2025 1:07 PM
servo	4/25/2025 1:07 PM
1.1Blink.sb3	1/31/2024 1:57 PM
1.2PWM.sb3	4/23/2025 8:20 AM
1.3Button.sb3	1/31/2024 1:57 PM
1.4Self-Locking-Button.sb3	1/31/2024 1:57 PM
1.5Lighting-System.sb3	1/31/2024 1:57 PM
2.1Photocell-sensor.sb3	1/31/2024 1:57 PM
2.2Light-Control-System.sb3	1/31/2024 1:57 PM



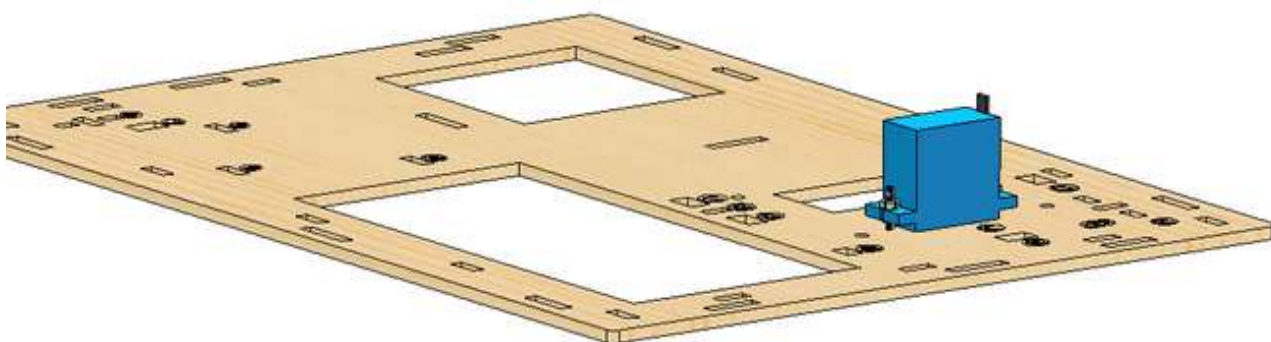
După setarea servo la un unghi specific, deconectați-l de la placa de dezvoltare pentru instalare.

4.3 Instalați servo

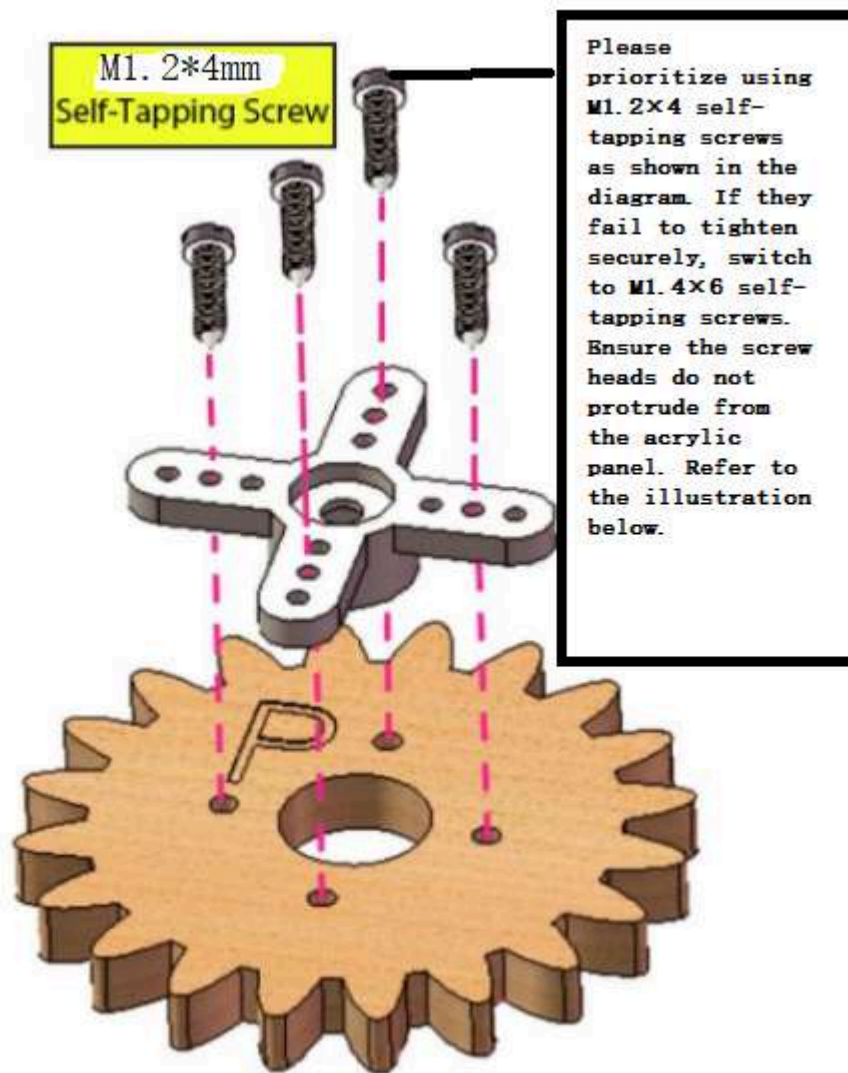


4.4

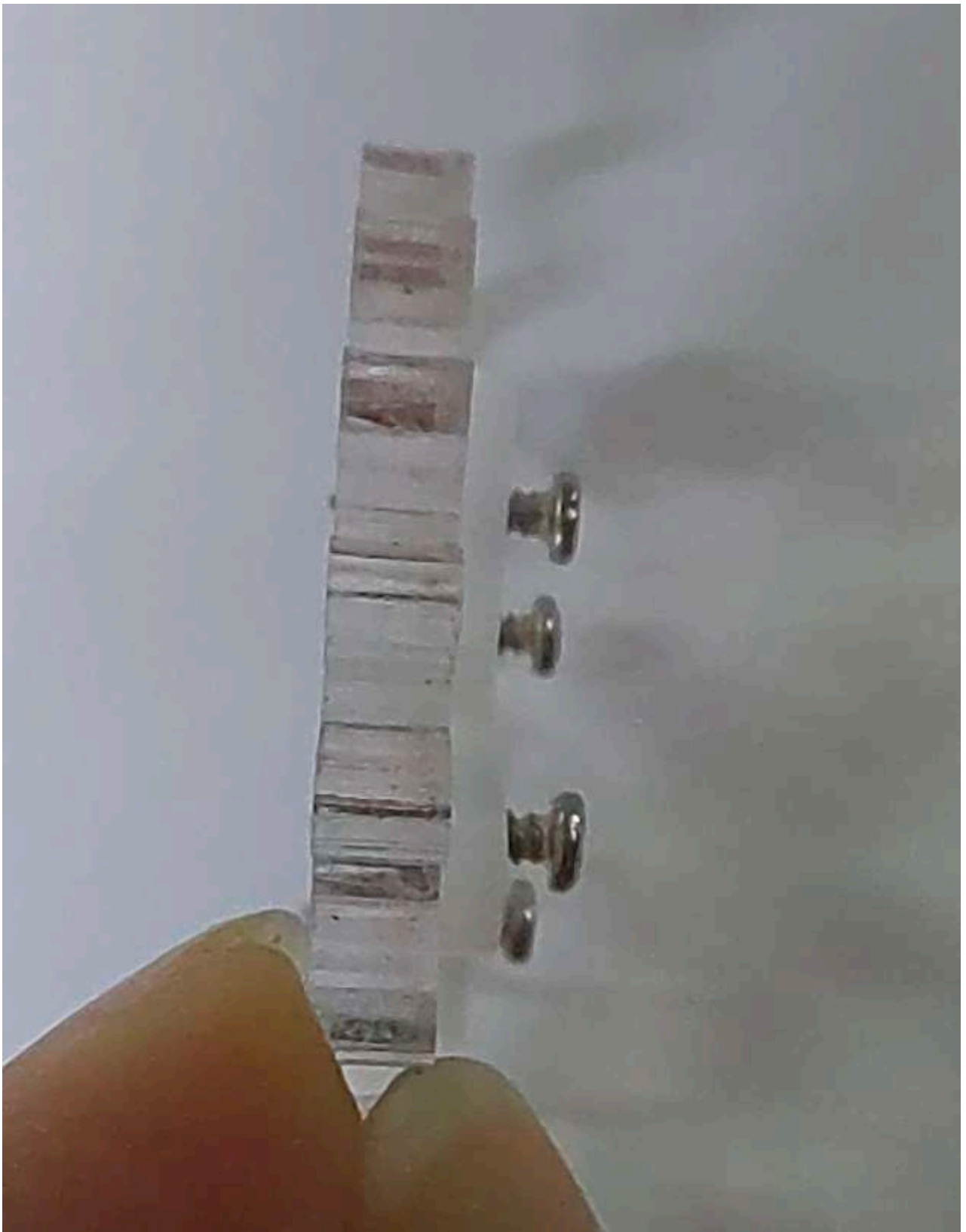
Notă: Șuruburile trebuie strânse pentru a menține servo-ul stabil, în caz contrar, ușa se poate bloca



4.5



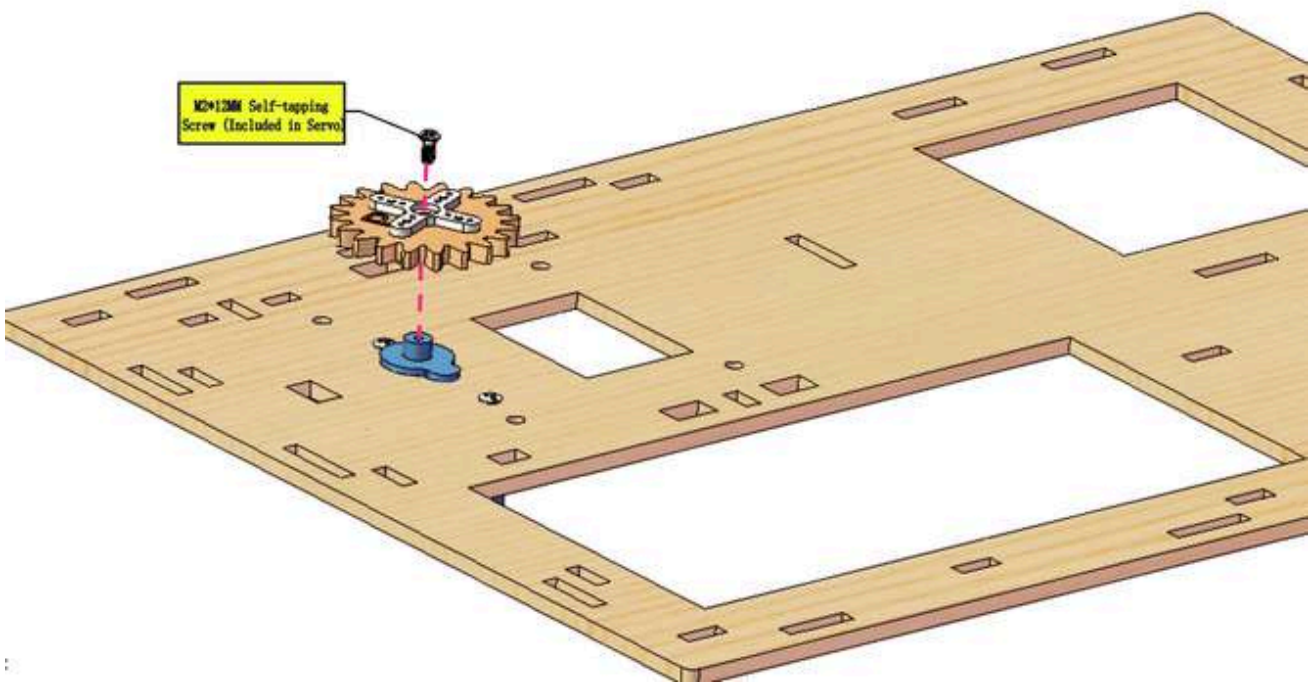
Instalați șuruburile autofiletante M1.4*6MM așa cum se arată mai jos



4.6

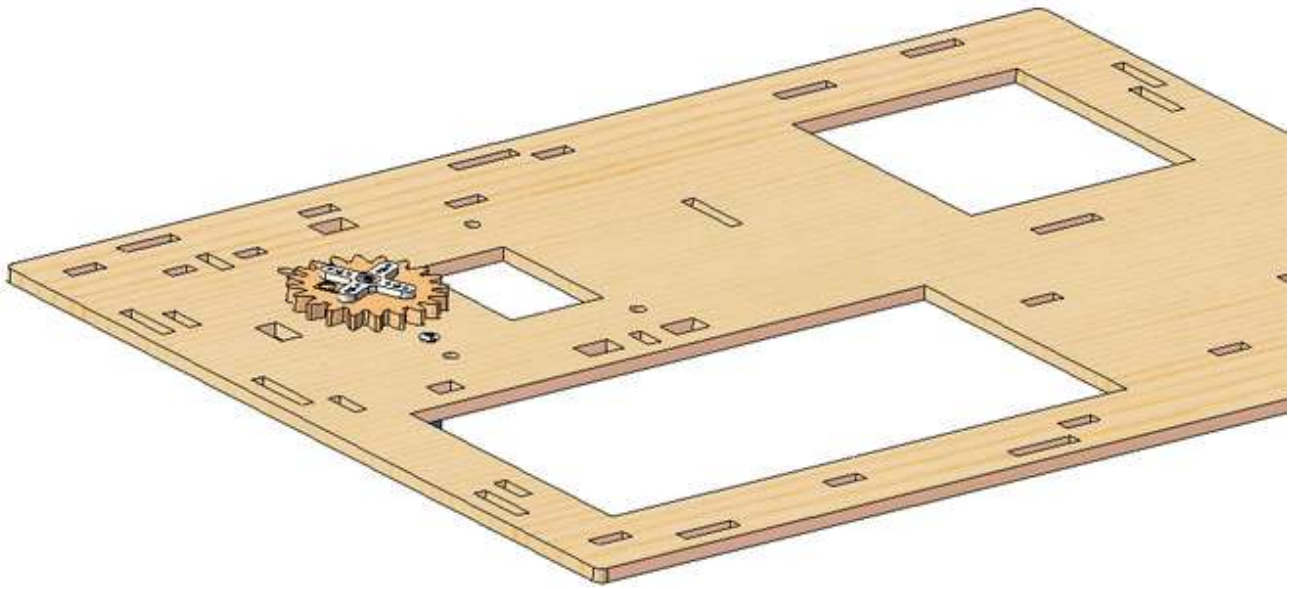


4.7



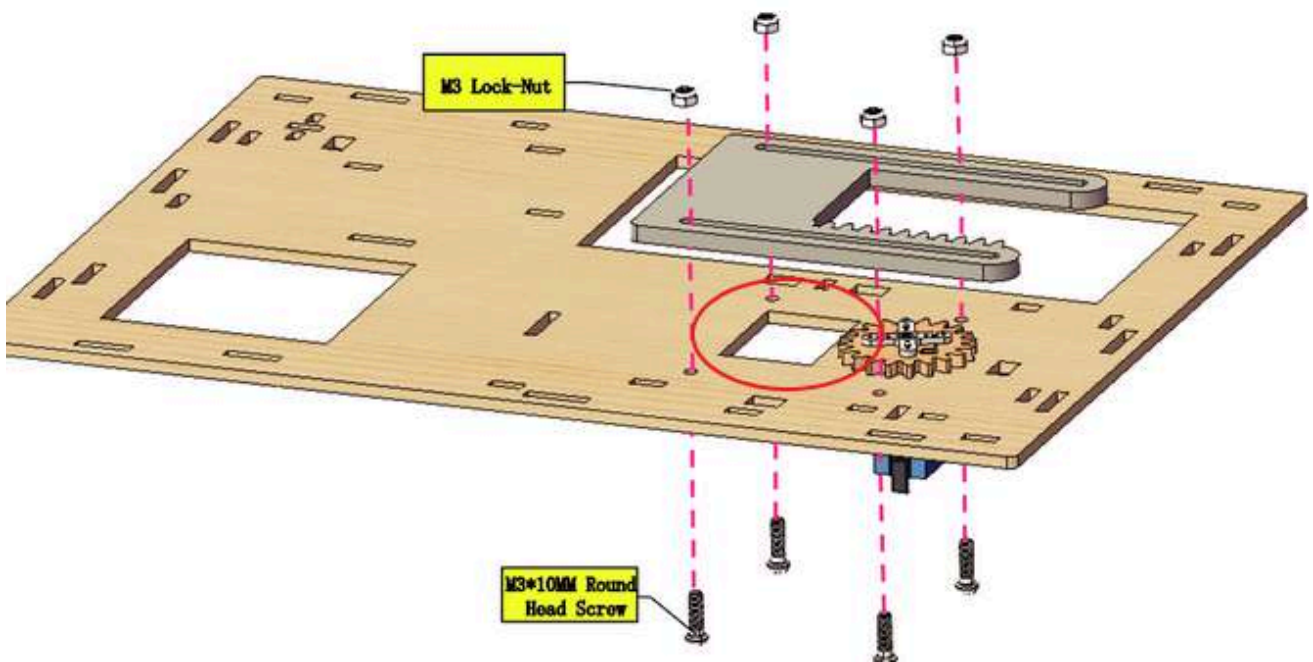
4.8

Nu rotiți angrenajul după ce este instalat pe servo. Dacă aveți Deja rotiți treapta de viteză, va trebui să rereglați unghiul servo la 180°.

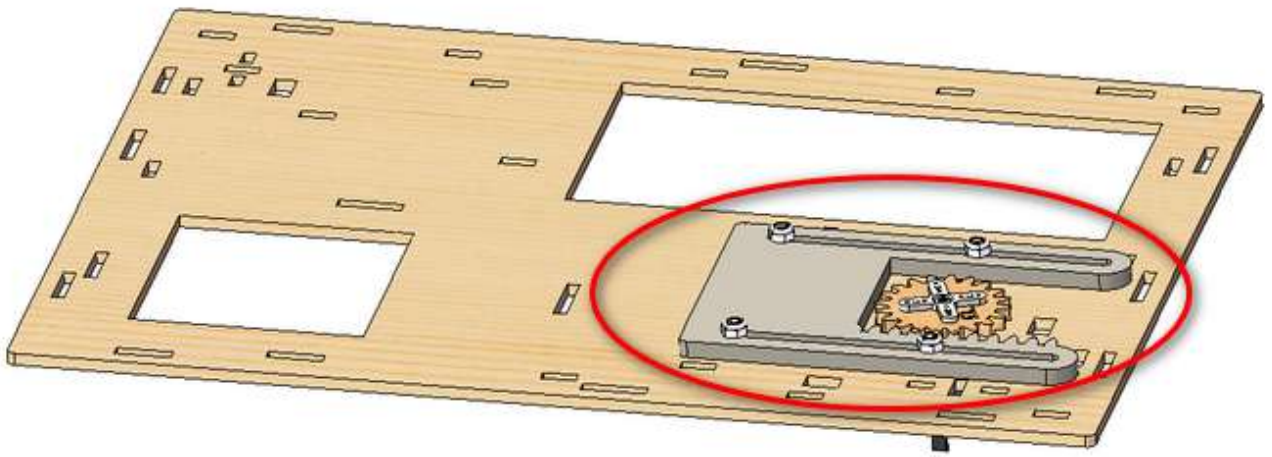


4.9

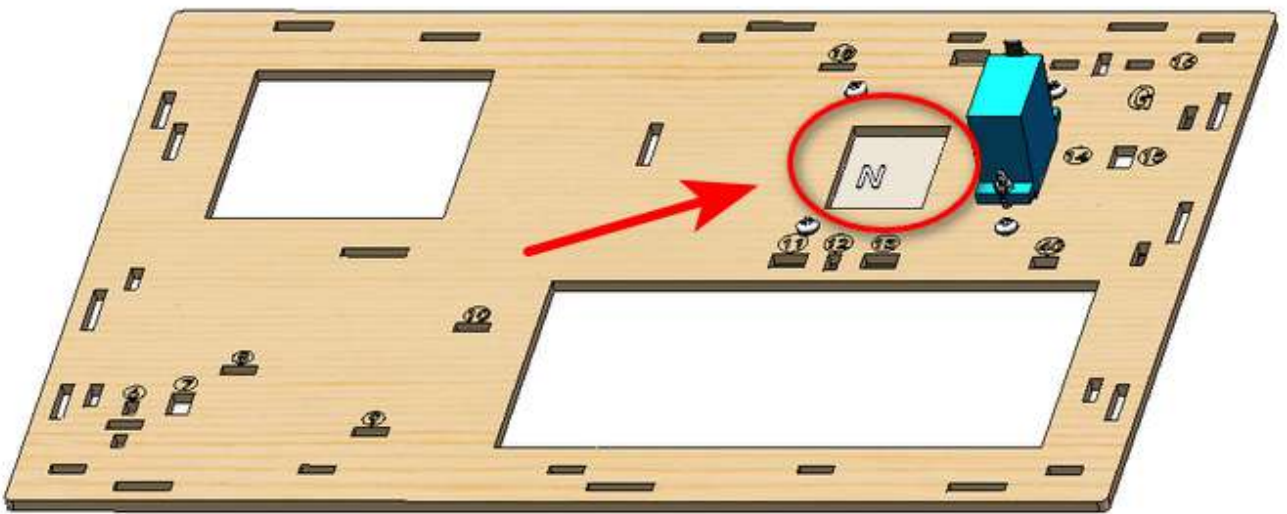
NOTĂ: Când instalați capacul cutiei de alimentare, deschiderea acestuia ar trebui să să fie complet închise.



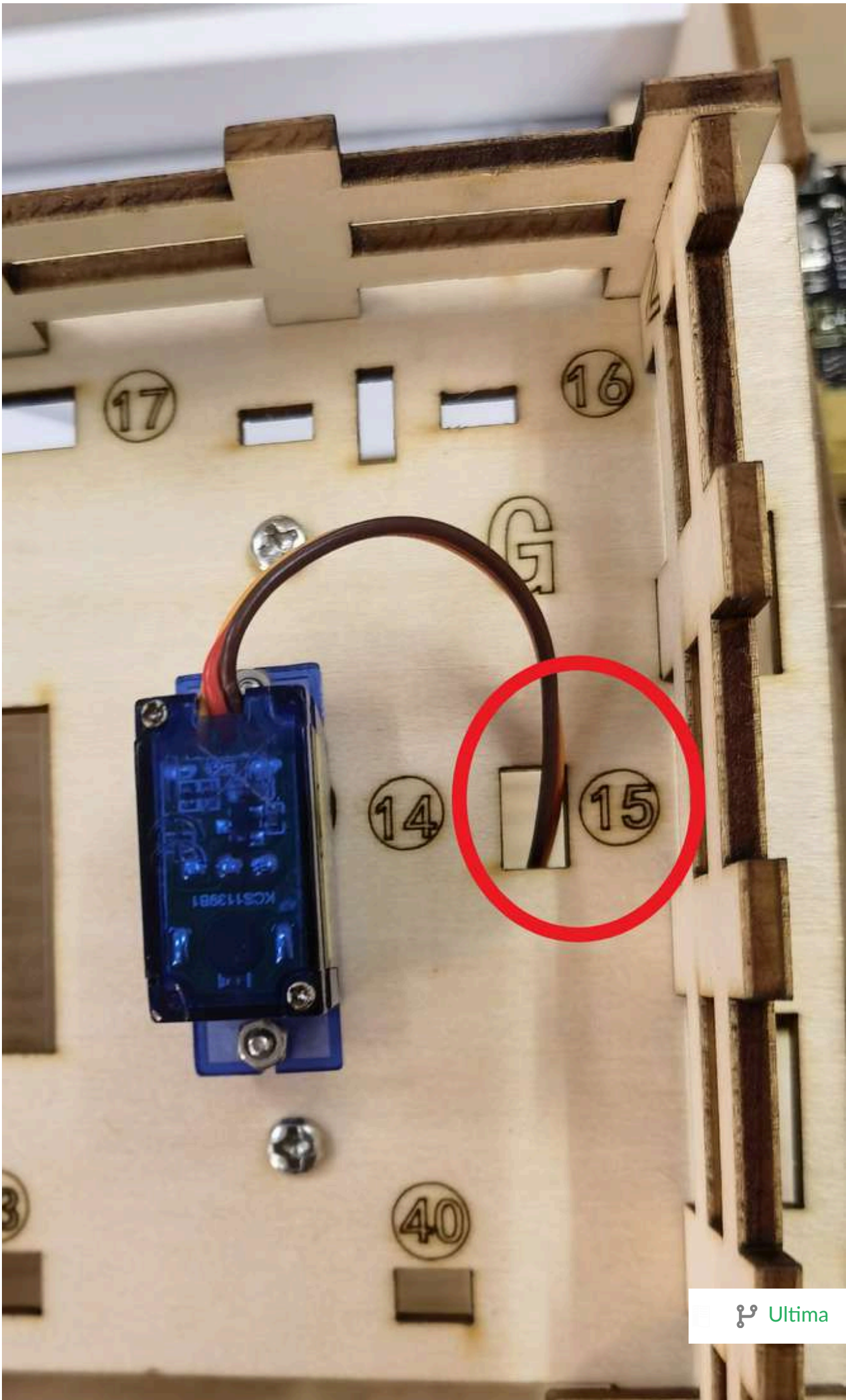
4.10



4.11

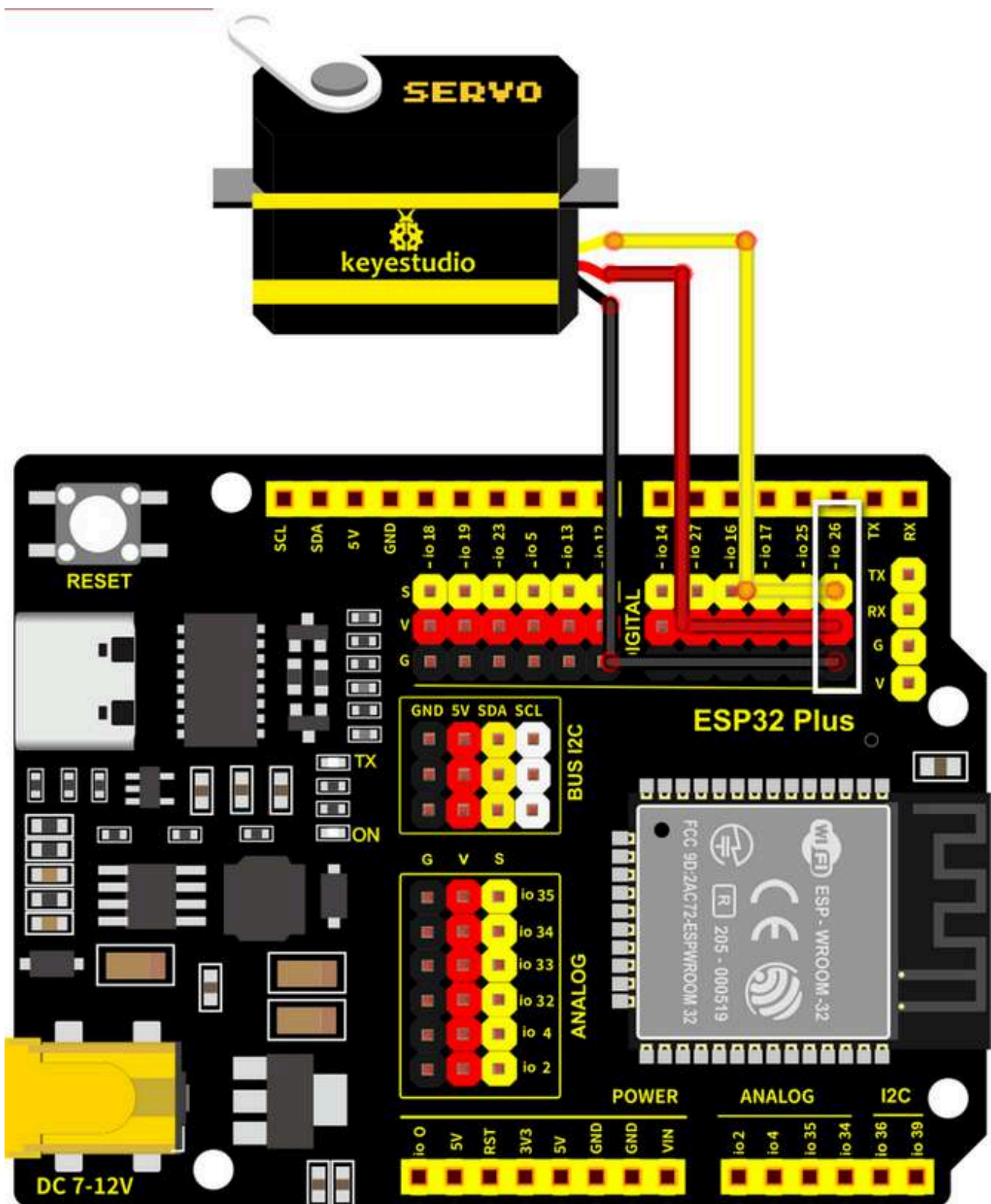


4.12



4.13 Testați ușa

1. Conectați Servo la pinul IO26 al plăcii ESP32. Conectați galben la S, roșu la V, negru la G.



2. Conectați 6 baterii AA la portul DC 7-12V al plăcii ESP32. (Bateriile nu sunt incluse în kit)

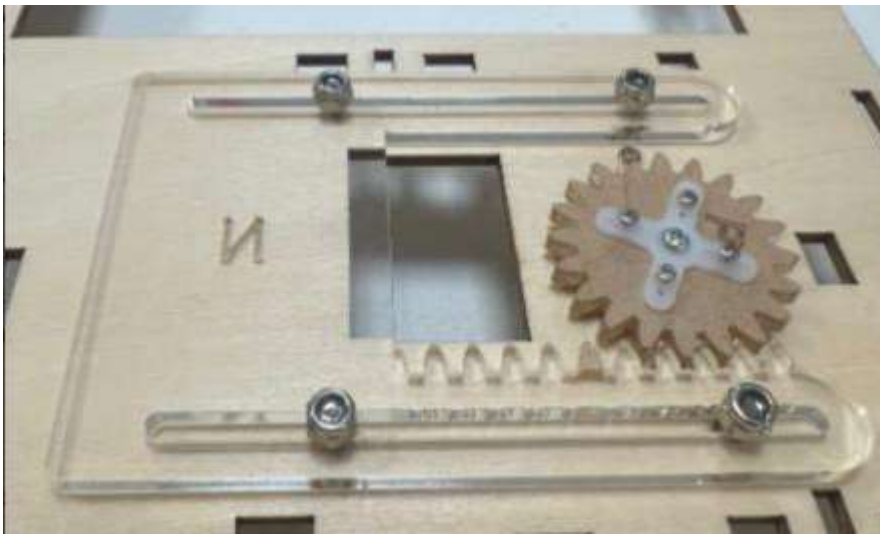


3. Încărcați codul de testare

A. Conectați placa ESP32 la computer cu cablul USB. Deschideți INO în folderul **6.1Servo.sb3** cu KidsBlock .

名称	修改日期
🔴 3.4Buzzer-Music.sb3	2024/1/31 13:57
🔴 3.5Alarm-System.sb3	2025/4/23 8:44
🔴 4.1Steam-Sensor.sb3	2024/1/31 13:57
🔴 4.2Rainwater-Detection-System.sb3	2024/1/31 13:57
🔴 6.1Servo.sb3	2024/1/31 13:57
🔴 6.2Ultrasonic-Sensors.sb3	2024/1/31 13:57
🔴 6.3Intelligent-Feeding-System.sb3	2024/1/31 13:57
🔴 7.1DHT11.sb3	2024/1/31 13:57
🔴 7.2LCD1602.sb3	2024/1/31 13:57
🔴 7.3Motor.sb3	2024/1/31 13:57
🔴 7.4Temperature-Control-System.sb3	2024/1/31 13:57
🔴 8.10-11 Humidity Sensor.sb3	2024/1/31 13:57

B.Încărcați codul



NOTĂ: După încărcarea codului, dacă ușa nu poate fi deschisă și închisă și servo este fierbinte, vă rugăm să opriți imediat alimentarea.

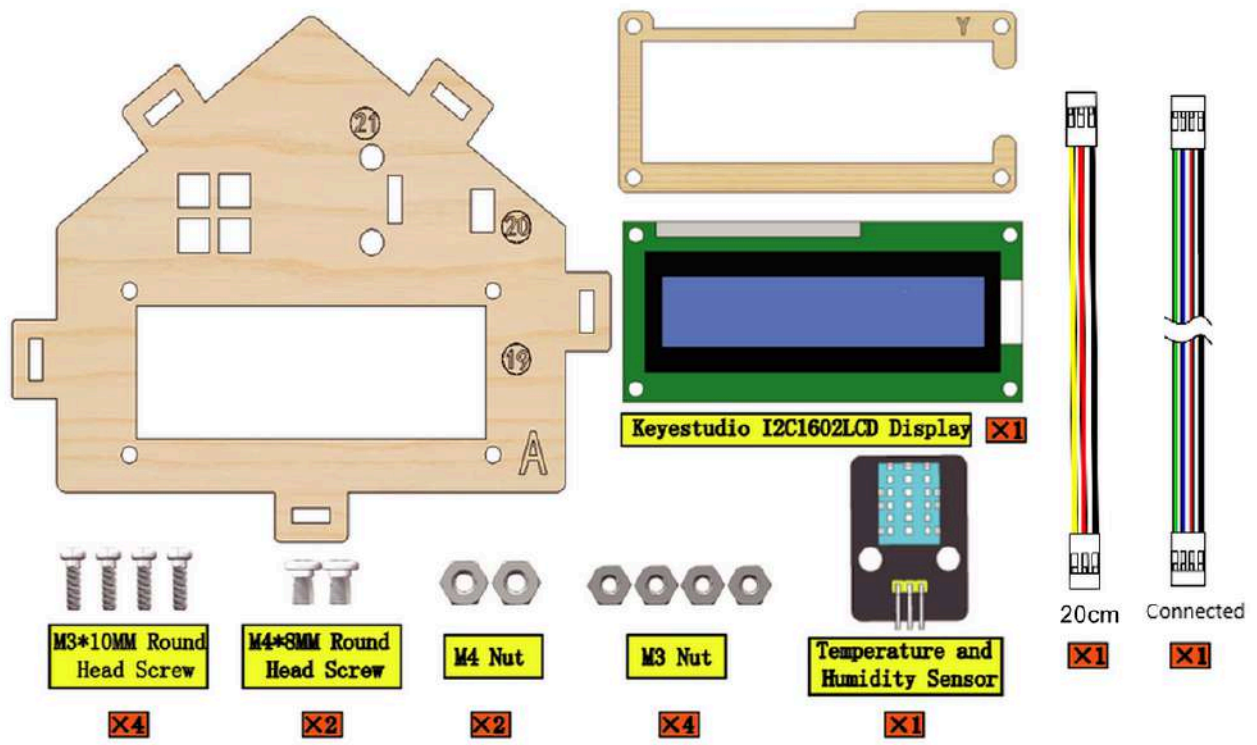
verificare:

1. Dacă ușa din plastic are puncte bune de contact și de forță cu structura angrenajului servo.
2. Dacă vârful șurubului de fixare de pe structura angrenajului servo este blocat cu carcasa de plastic a servo. Dacă da, vă rog Slăbiți puțin șurubul de fixare pentru a preveni contactul vârfului acestuia servo.

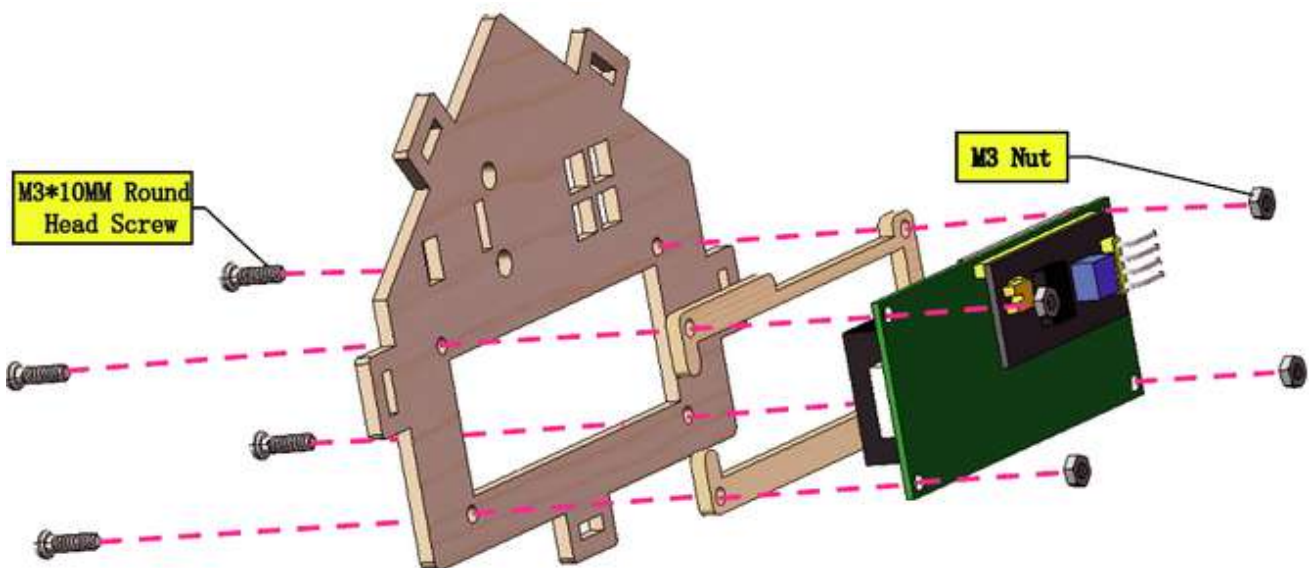


Pasul 5 Instalați afișajul LCD și senzorul DHT11

5.1 Componente necesare



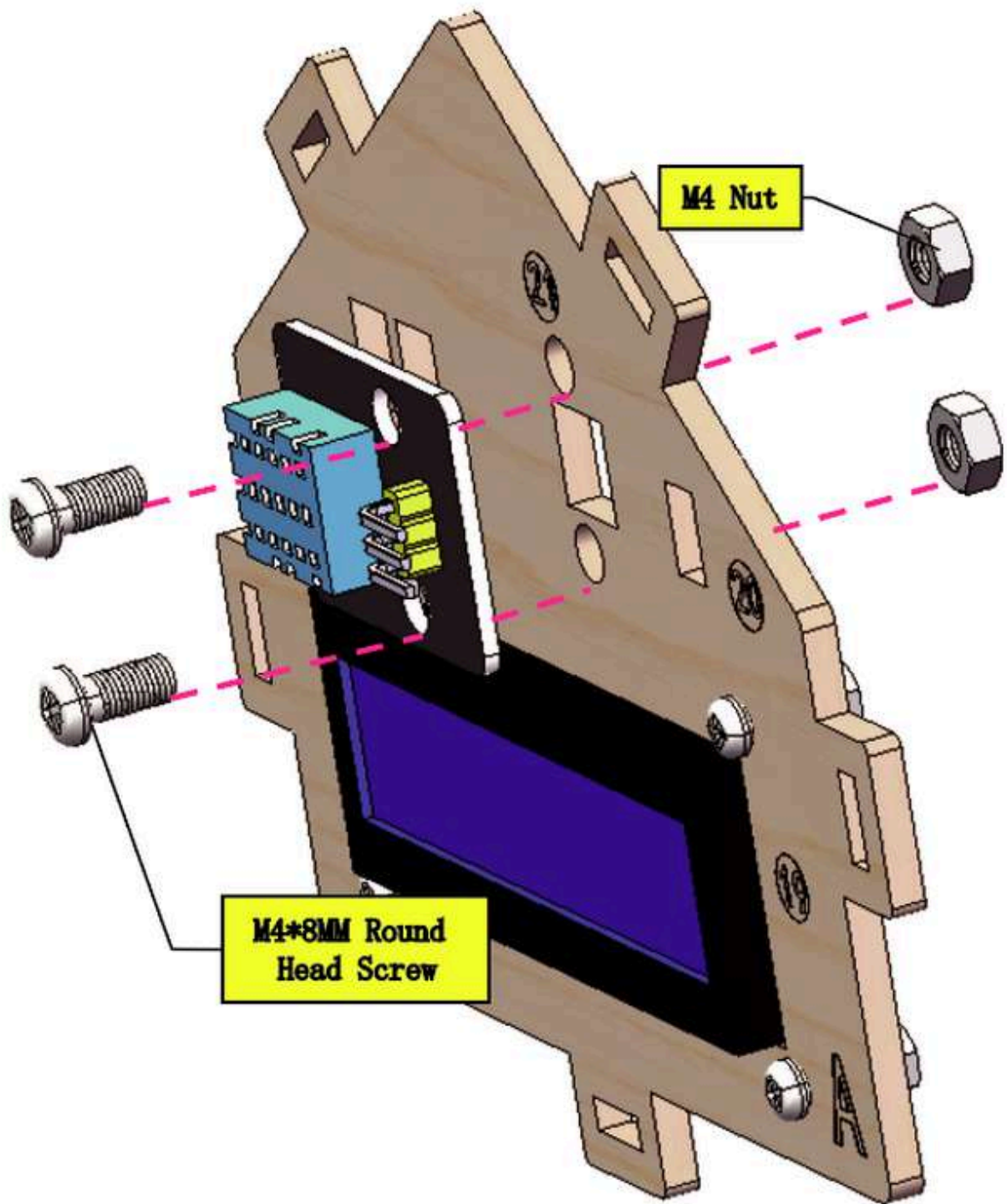
5.2



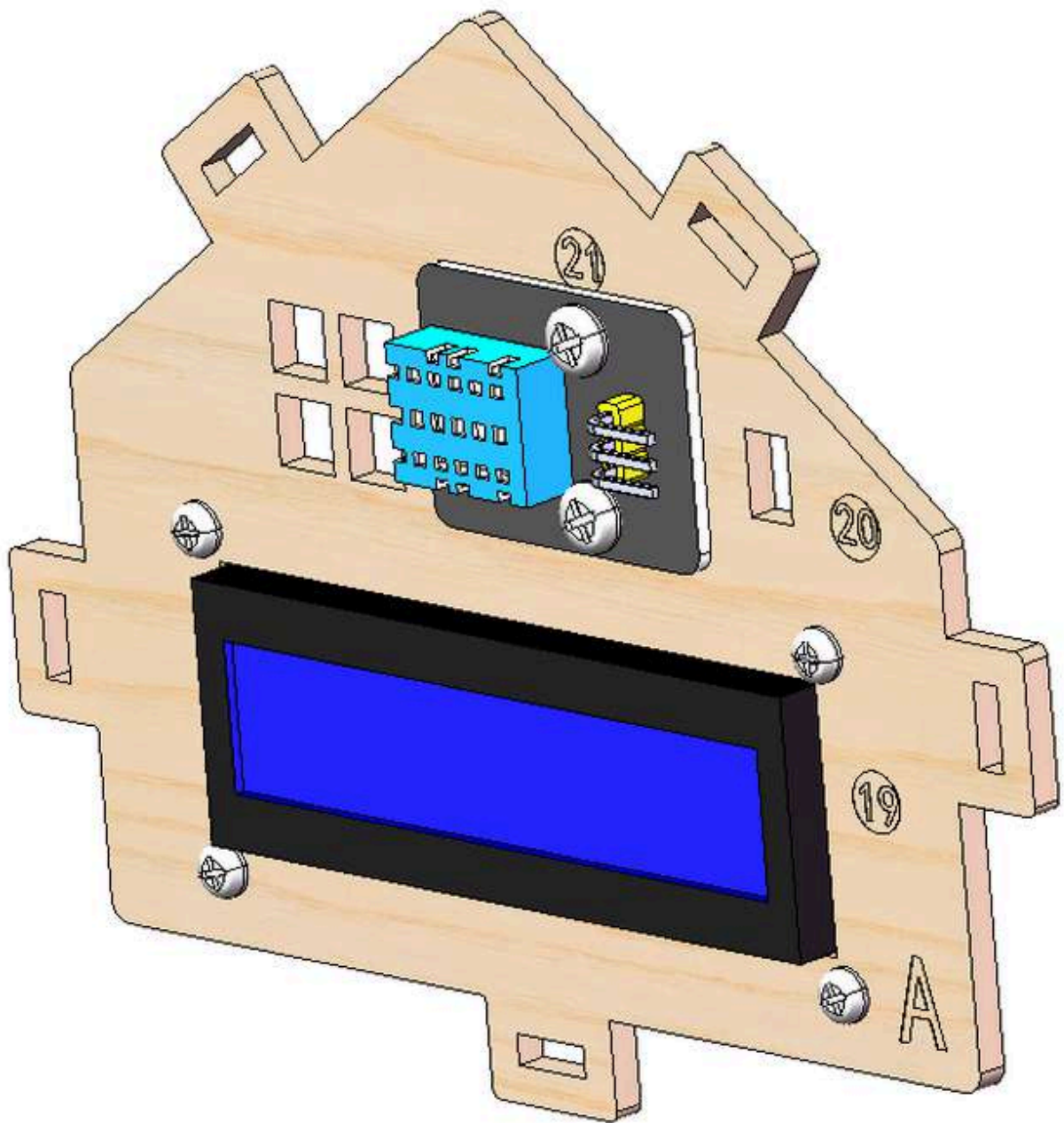
5.3



5.4



5.5 Prototype



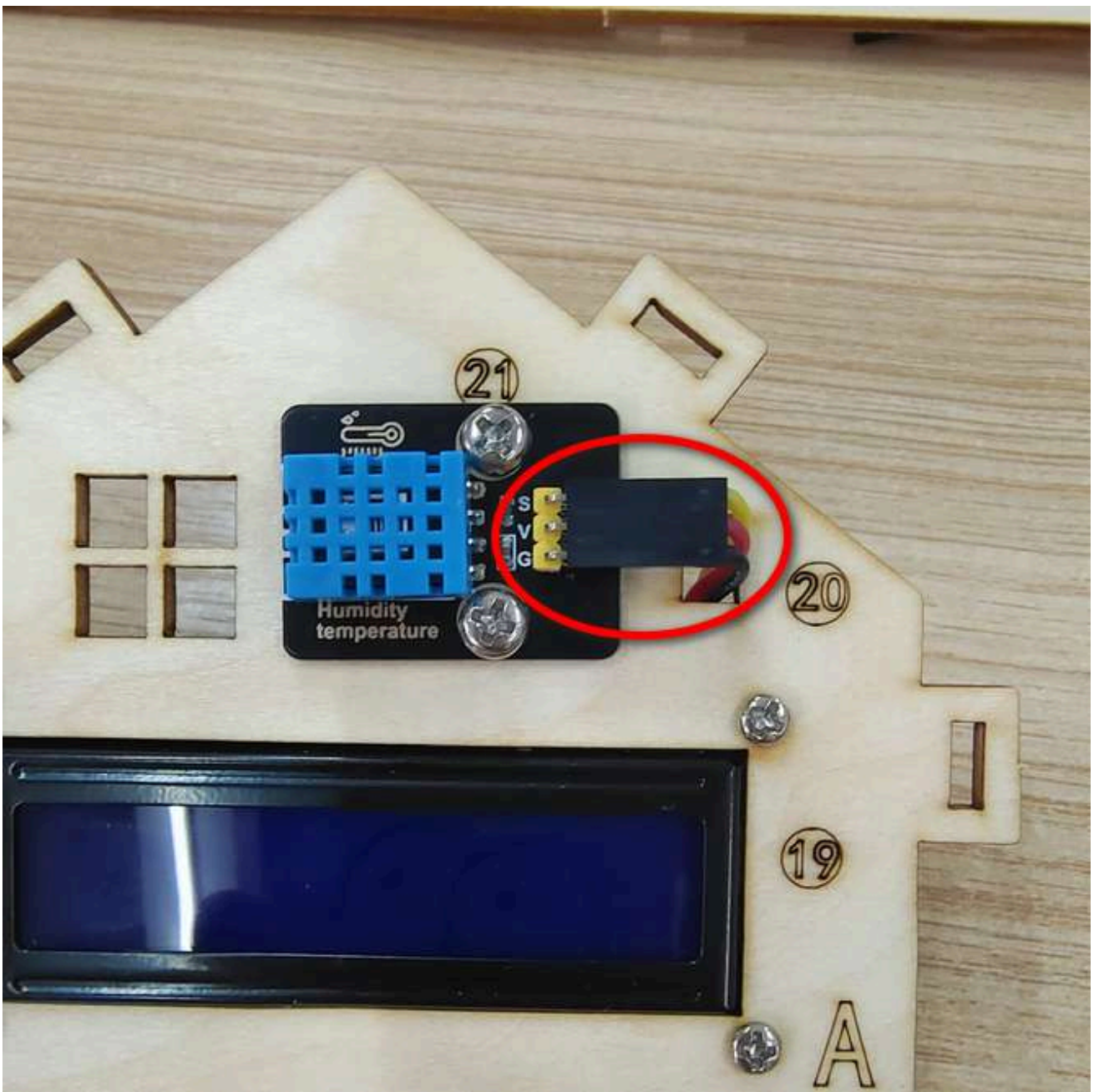
5.6 Wiring

Connect modules via Dupont wires.

Module	Wire
Temperature and Humidity Sensor	3PIN 20cm
LCD 1602	4PIN (Black-Red-Blue-Green)

Acordați atenție culorii firului Dupont:

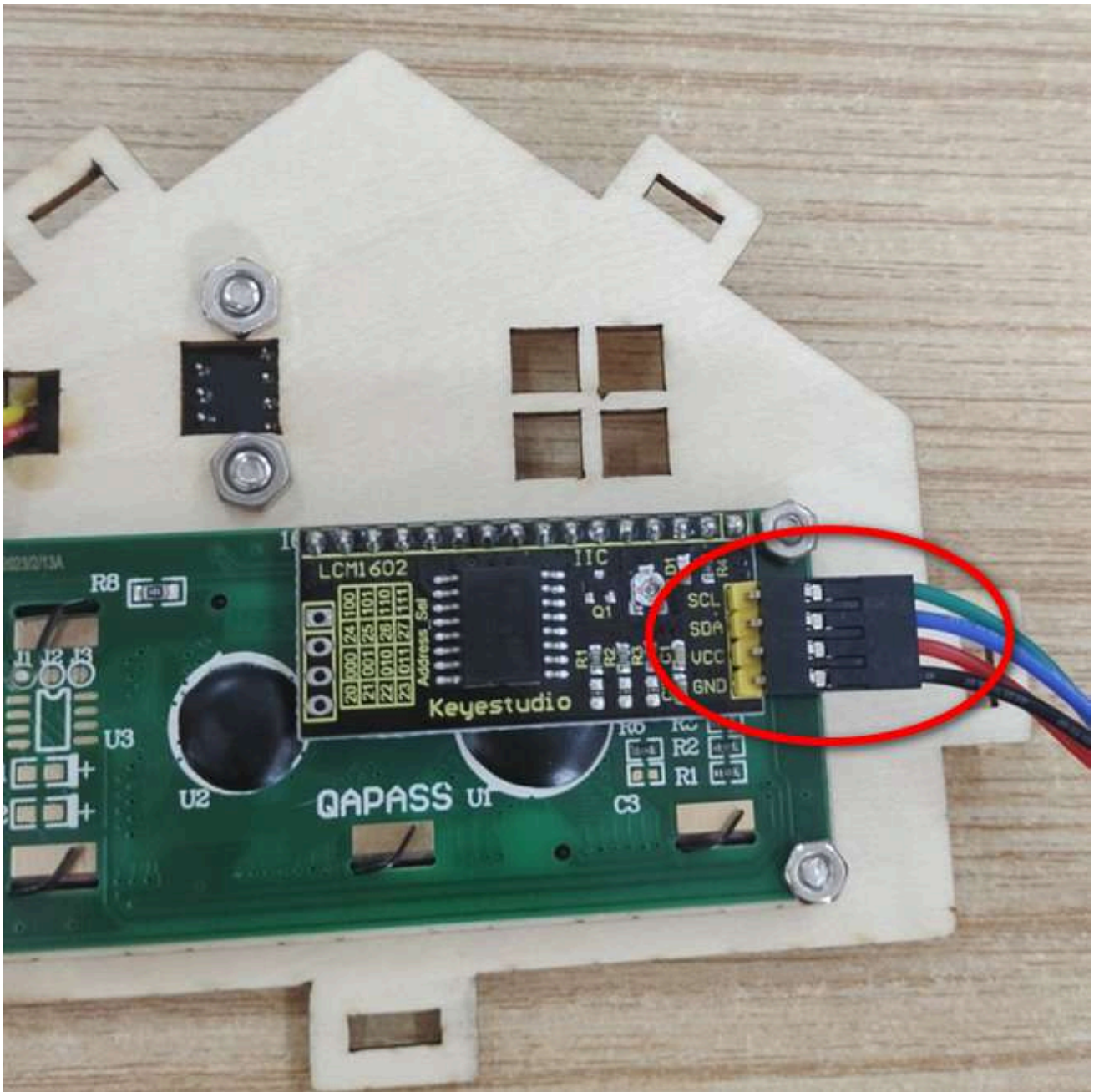
Pentru senzorul de temperatură și umiditate, conectați galben la S, roșu la V, negru la G.



5.7

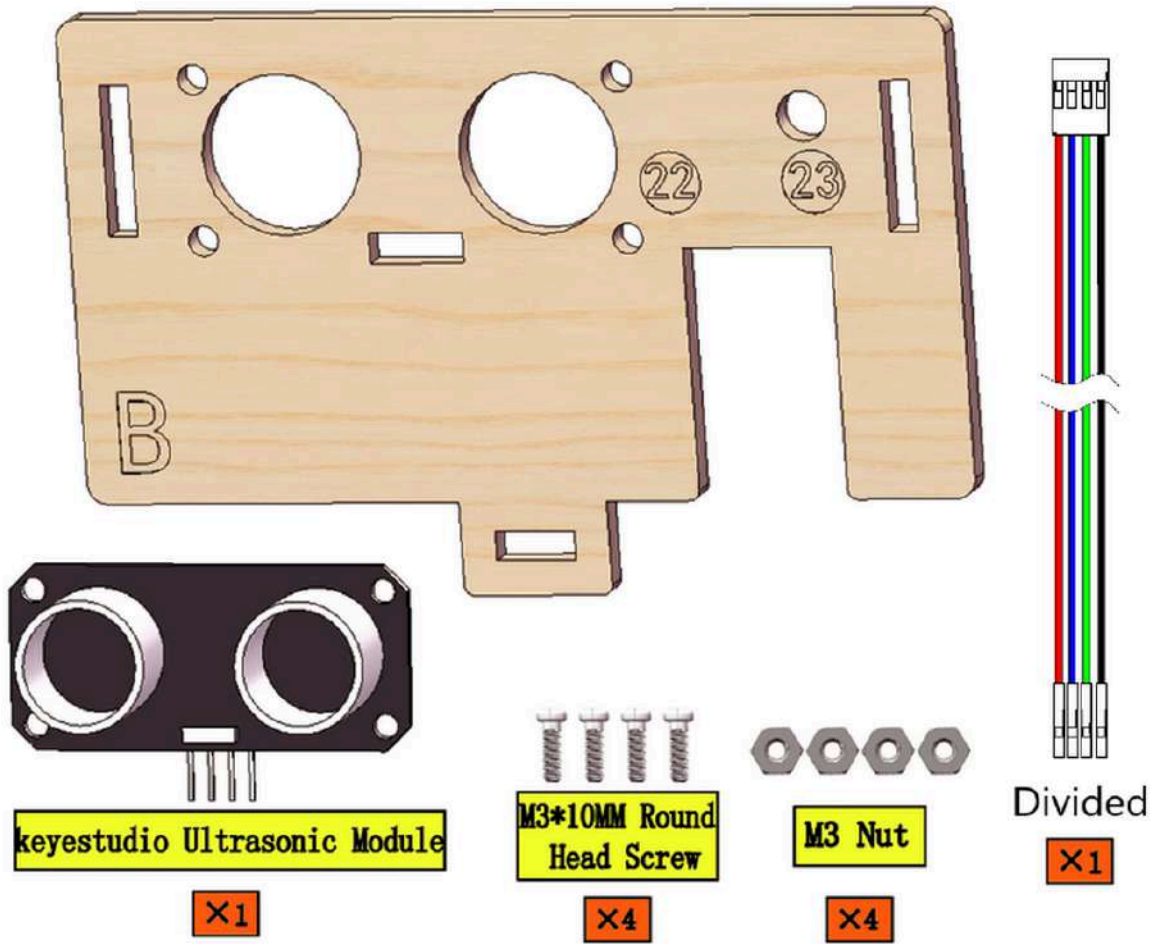
Modul	Sârmă
LCD 1602	4 pini (negru-roșu-albastru-verde)

Pentru afișajul LCD, conectați verde la SCL, albastru la SDA, roșu la VCC, negru la GND.

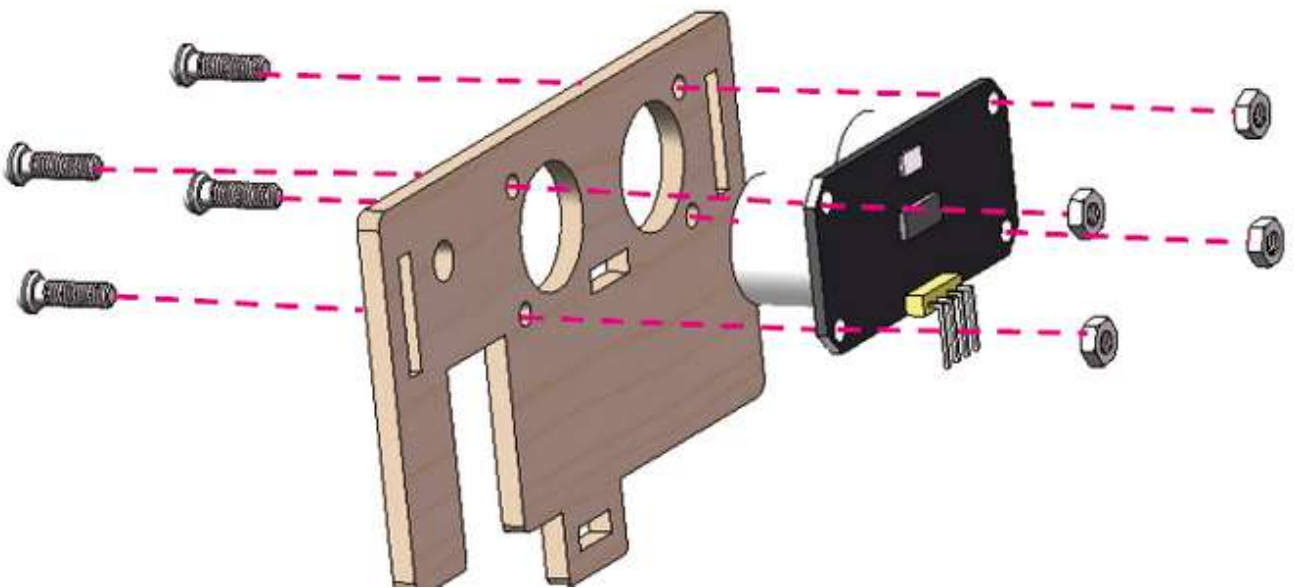


Pasul 6 Instalați modulul cu ultrasunete

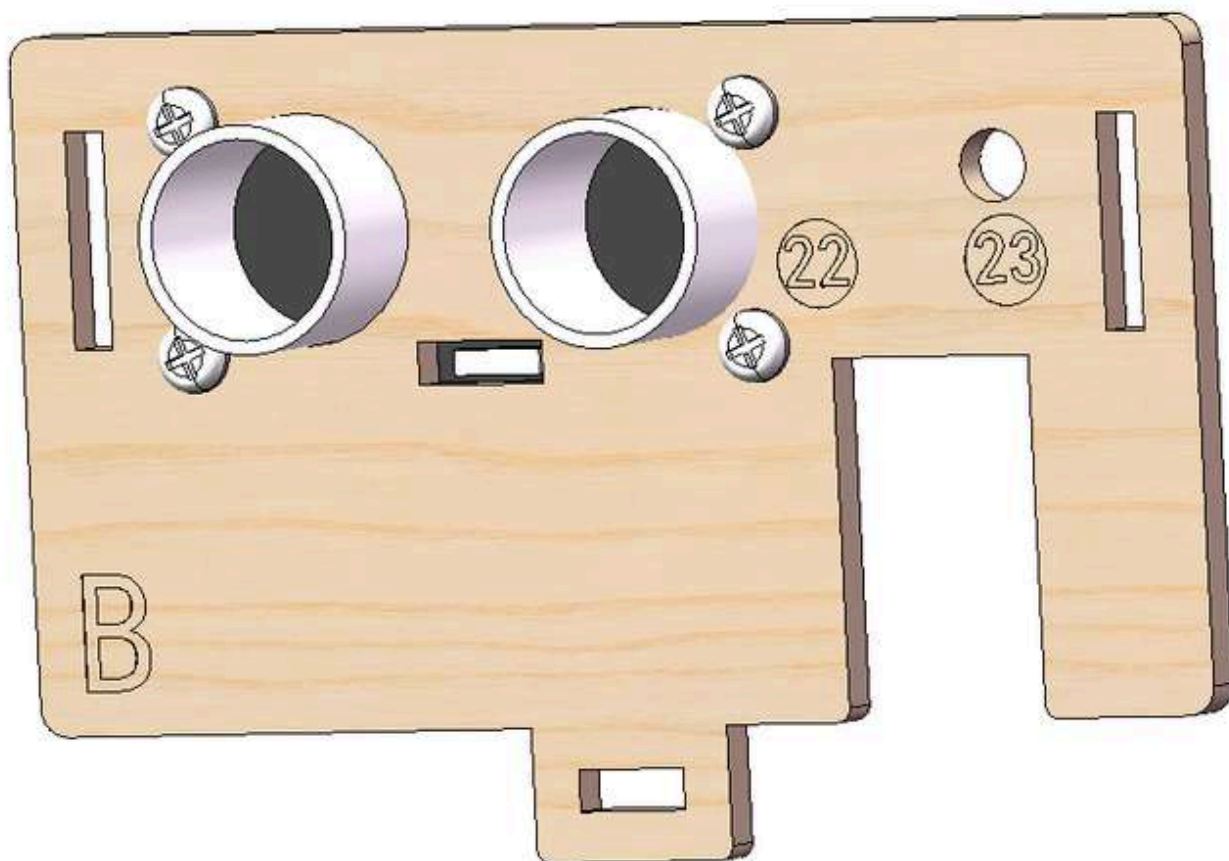
6.1 Componente necesare



6.2



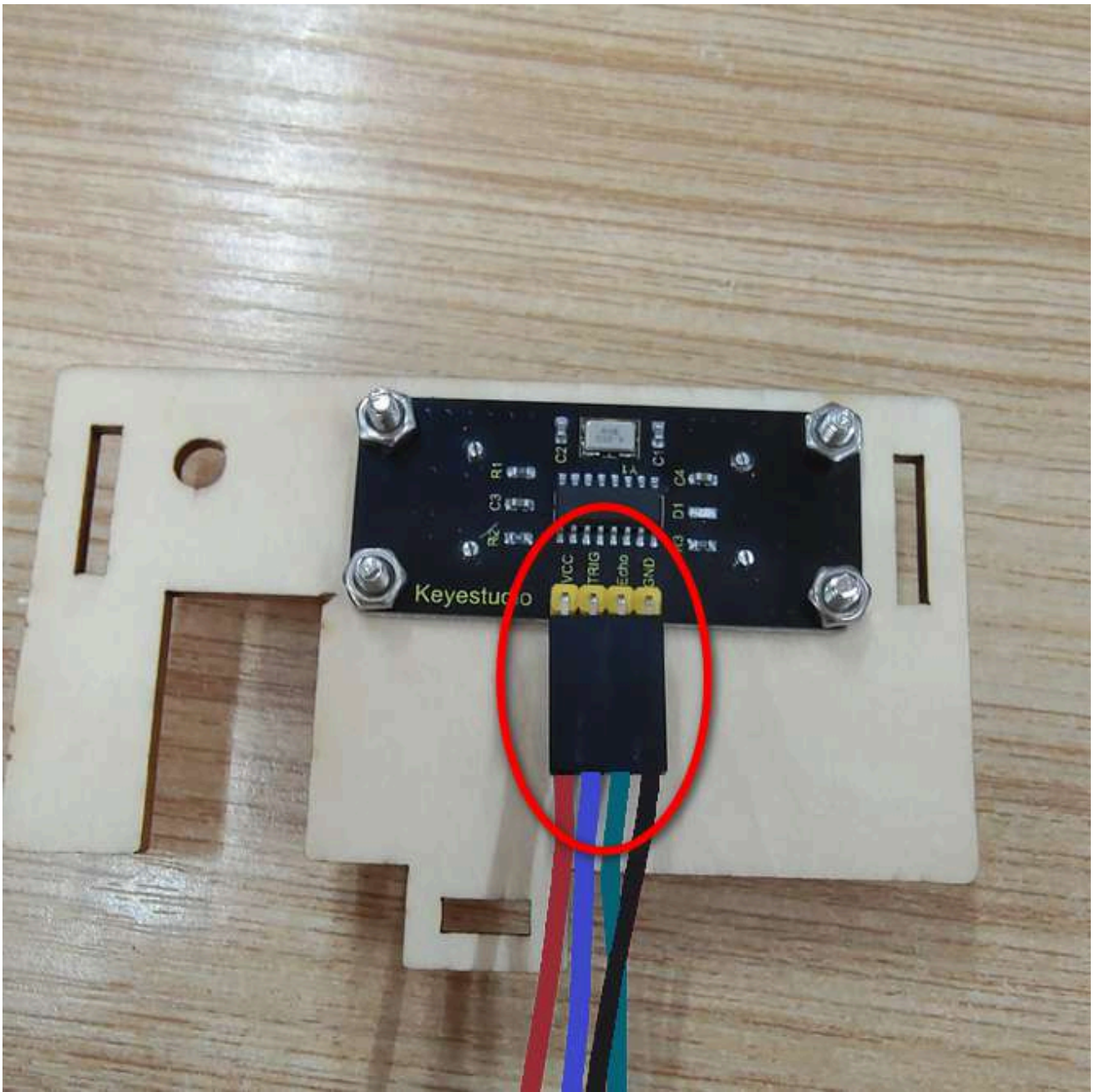
6.3



6.4 Cablare

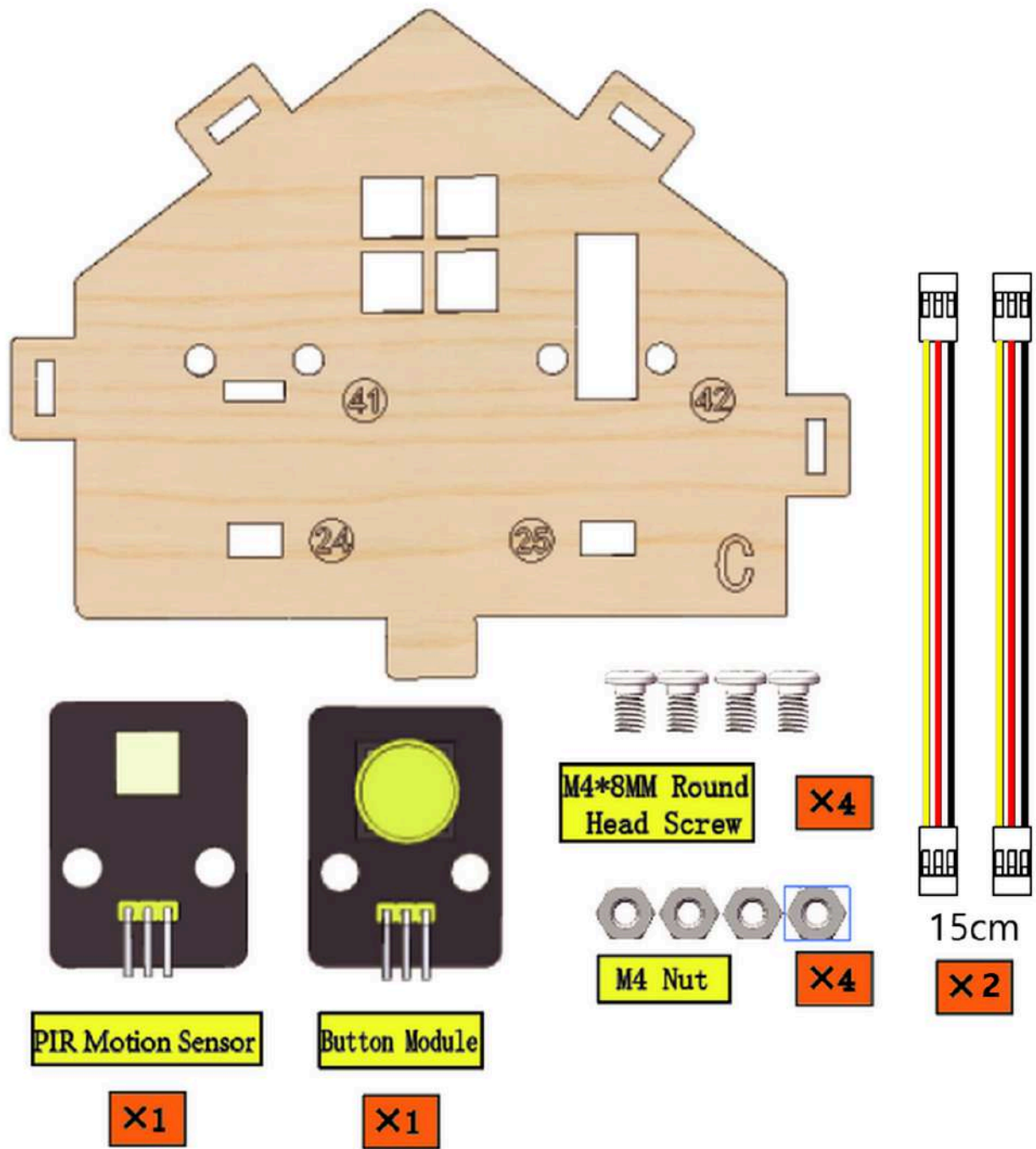
Modul	Sârmă
Modul cu ultrasunete	4 pini (negru-verde-albastru-roșu)

Acordați atenție culorii firului Dupont: Pentru ultrasunete modul, conectați albastru la TRIG, verde la ECHO, roșu la VCC, negru la GND.

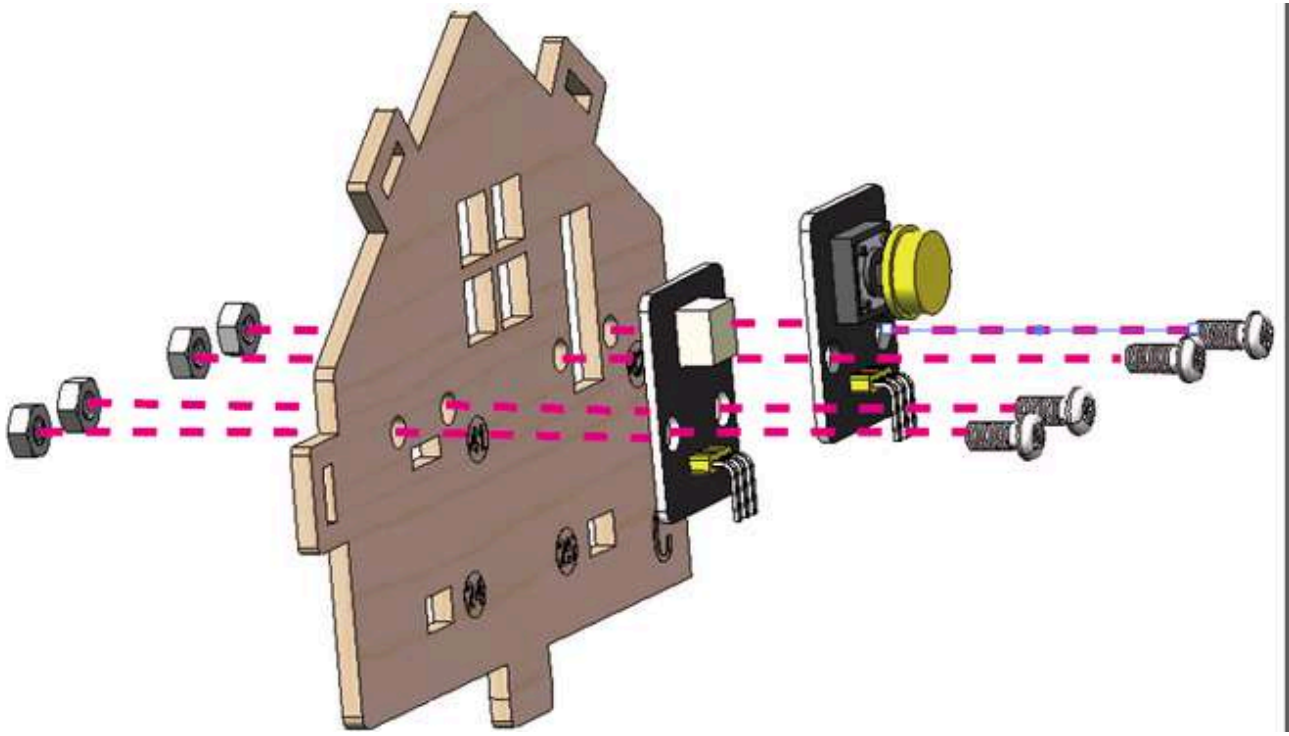


Pasul 7 Instalați senzorul de mișcare PIR și modulul buton

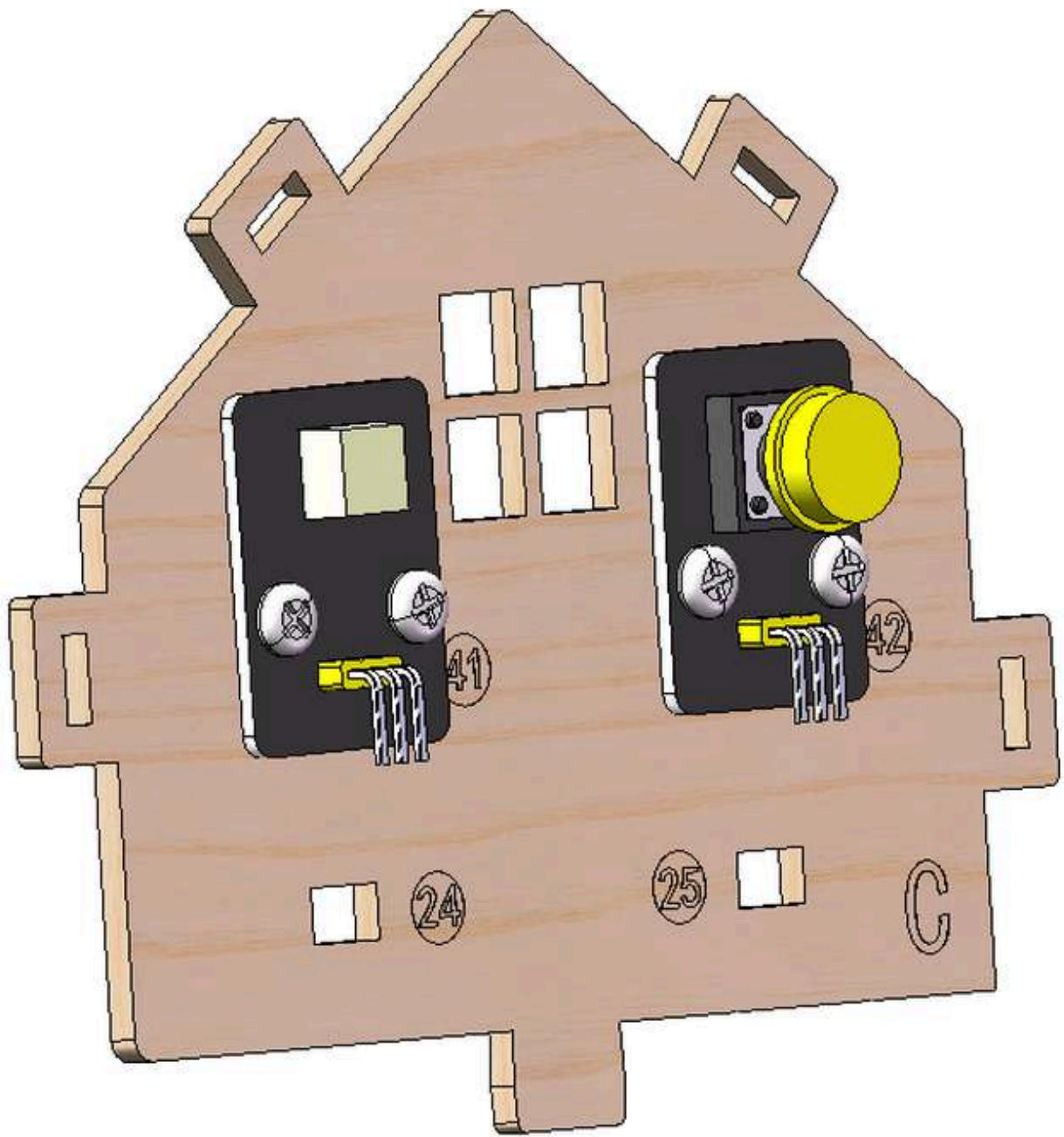
7.1 Componente necesare



7.2



7.3

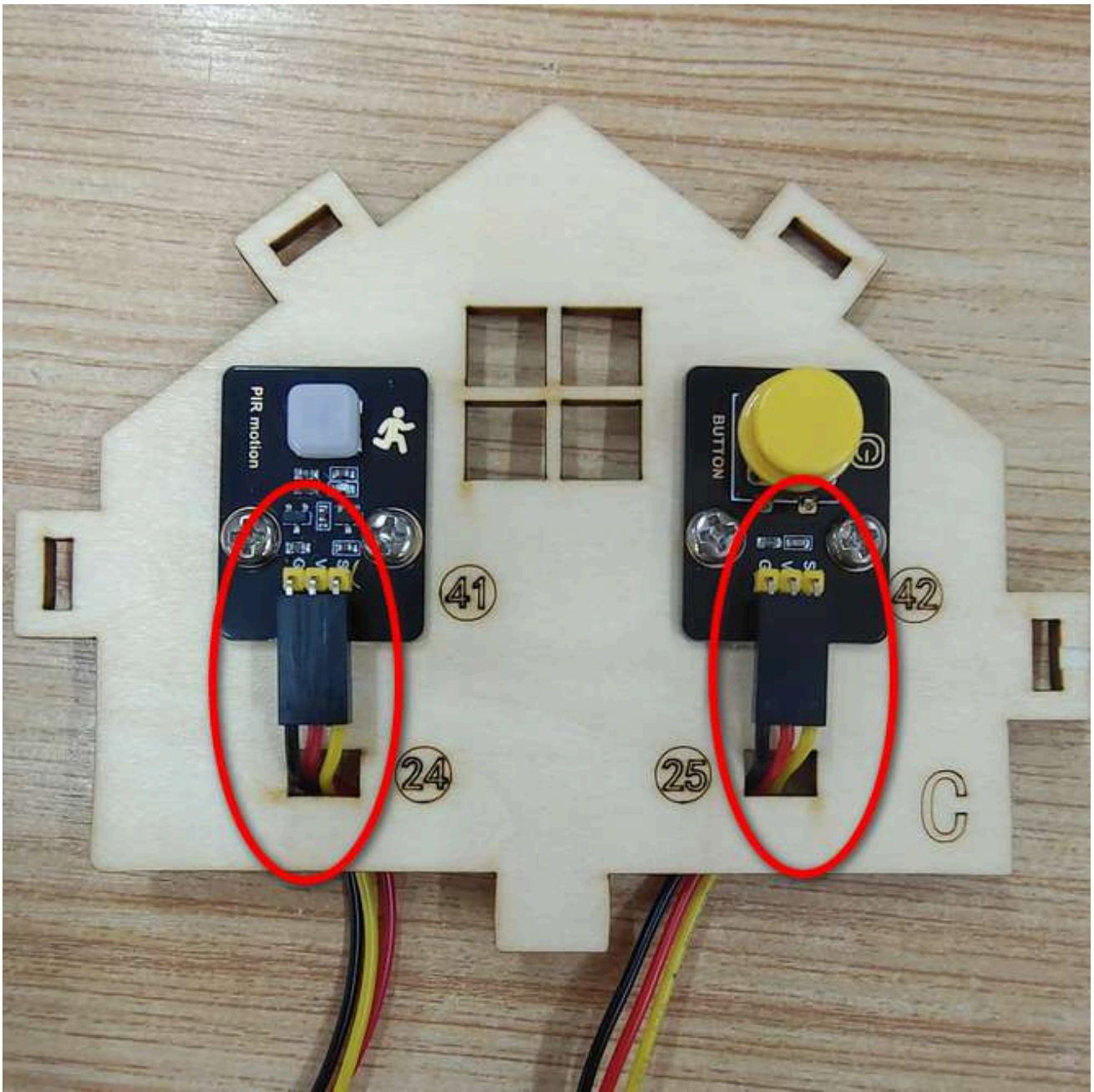


7.4 Wiring

Connect modules via Dupont wires.

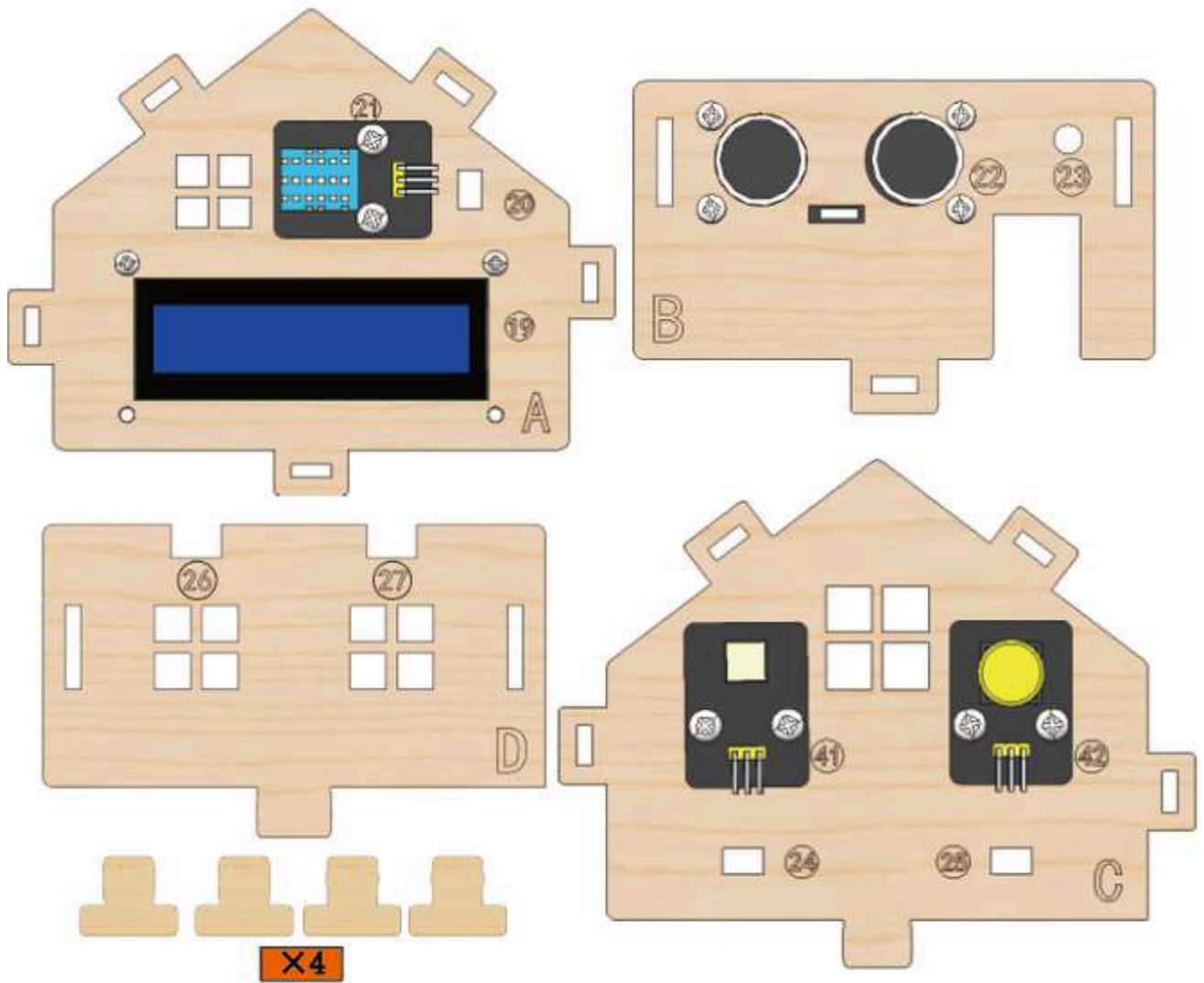
Module	Wire
PIR Motion Sensor	3PIN 15cm
Button Module	3PIN 15cm

Pay attention to the color of the Dupont wire: Connect yellow to S, red to V, black to G.

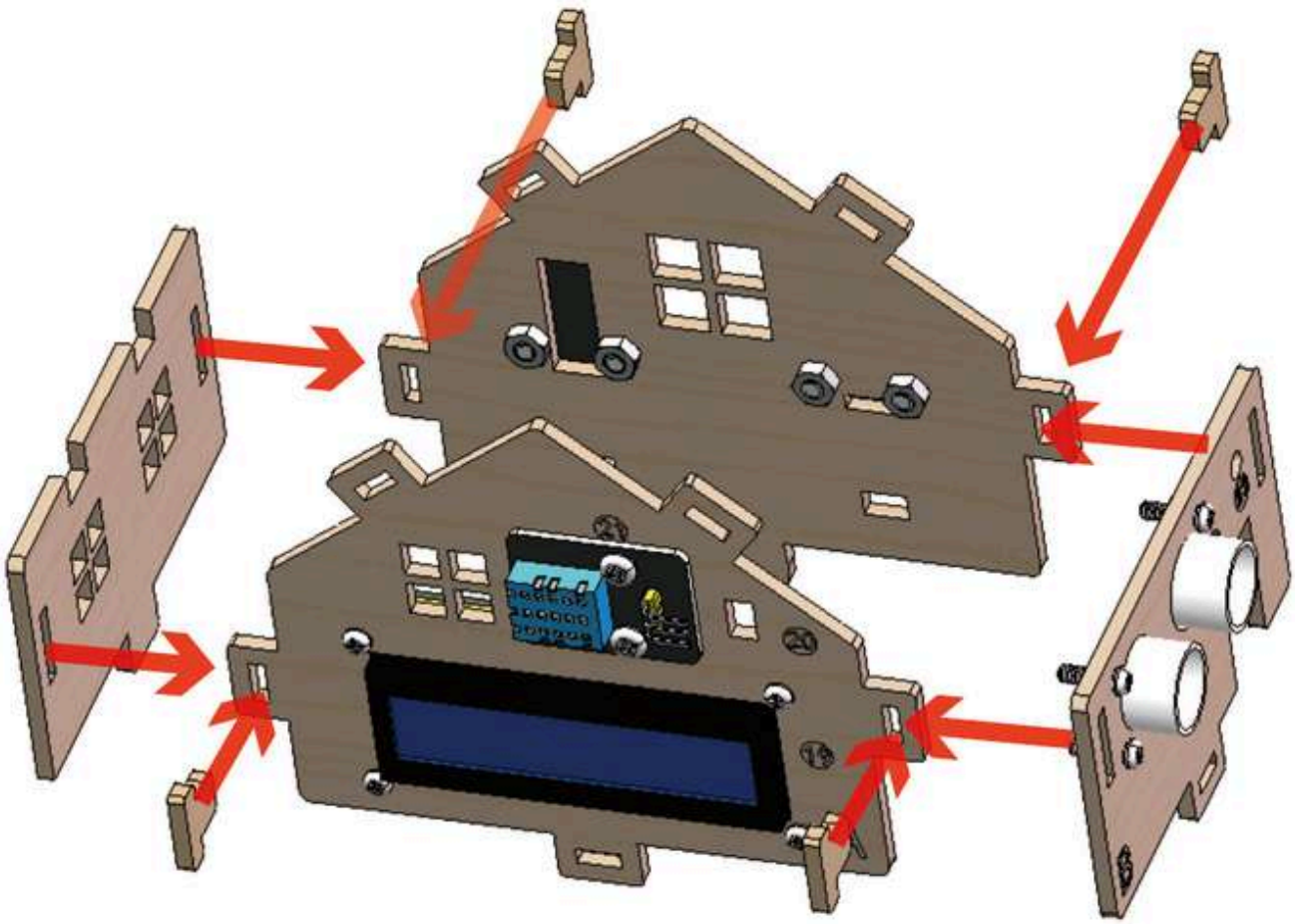


Pasul 8 Instalați pereții casei

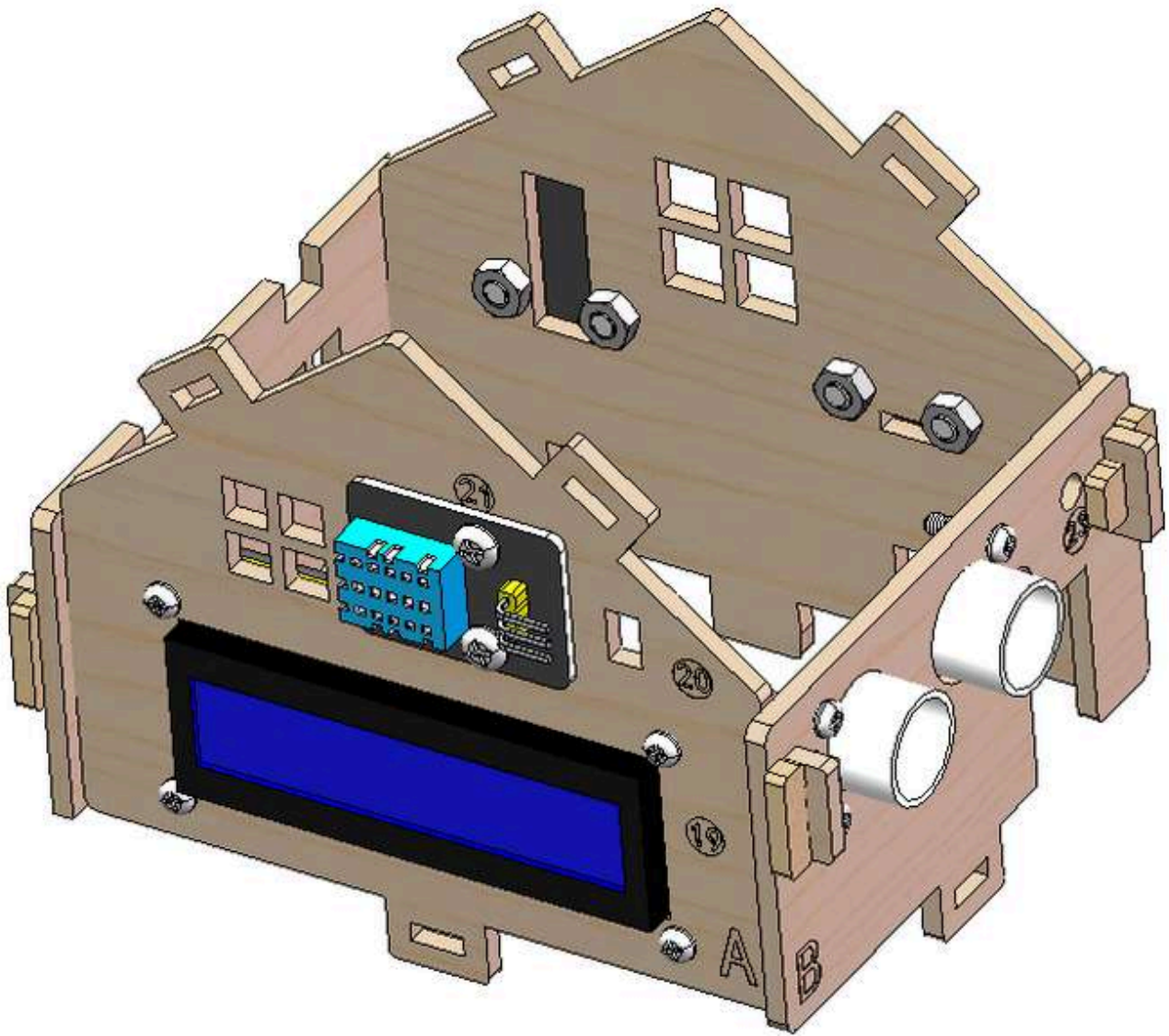
8.1 Componente necesare



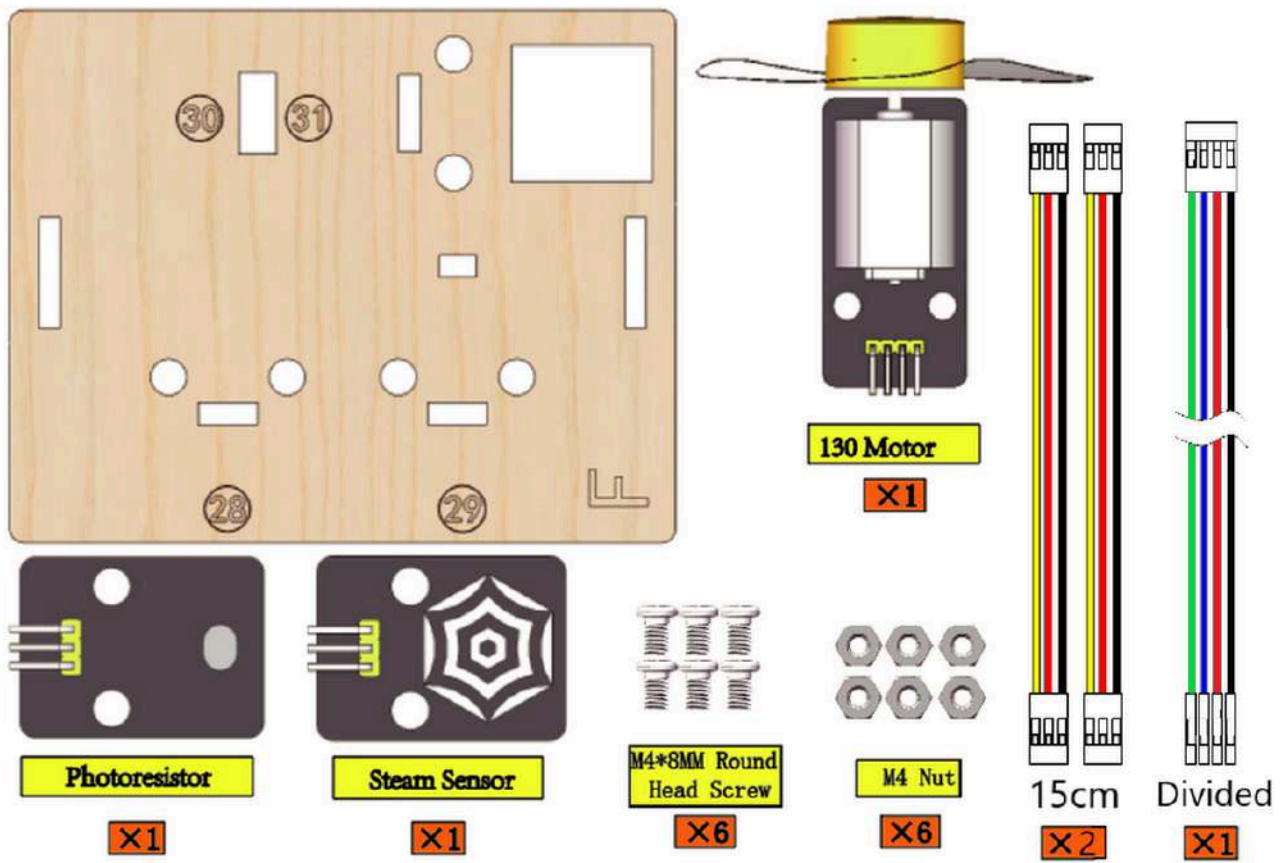
8.2



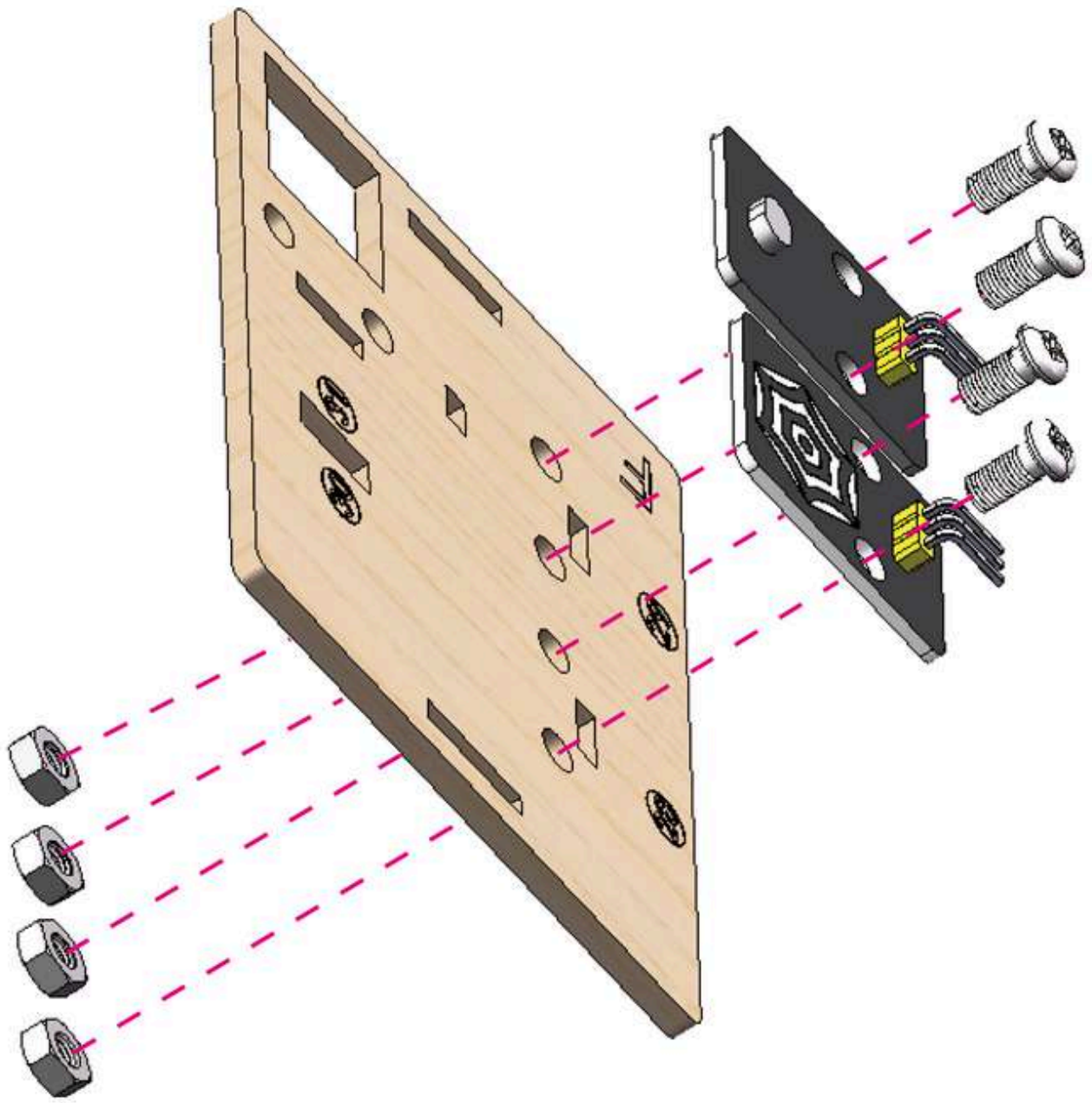
8.3



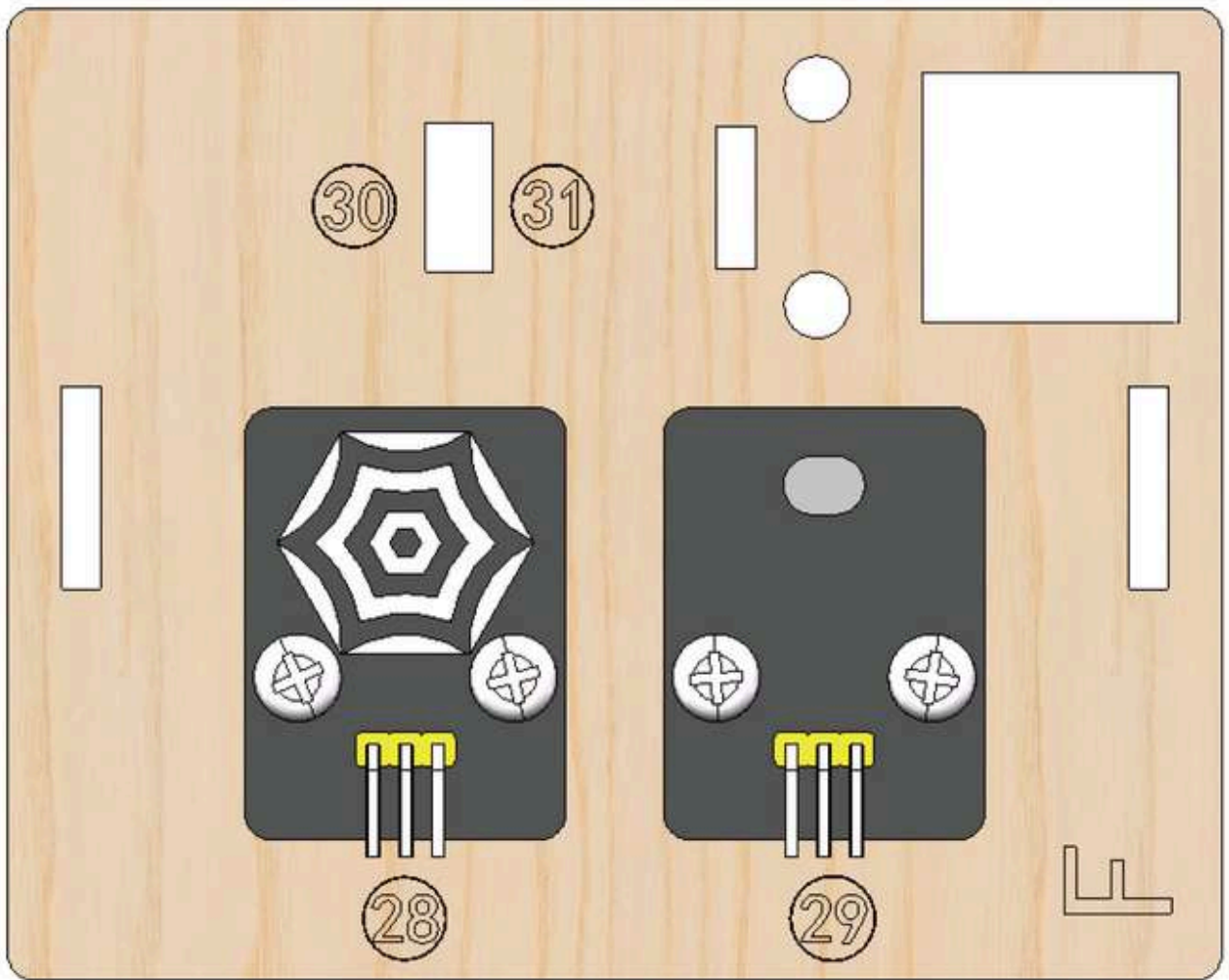
8.4



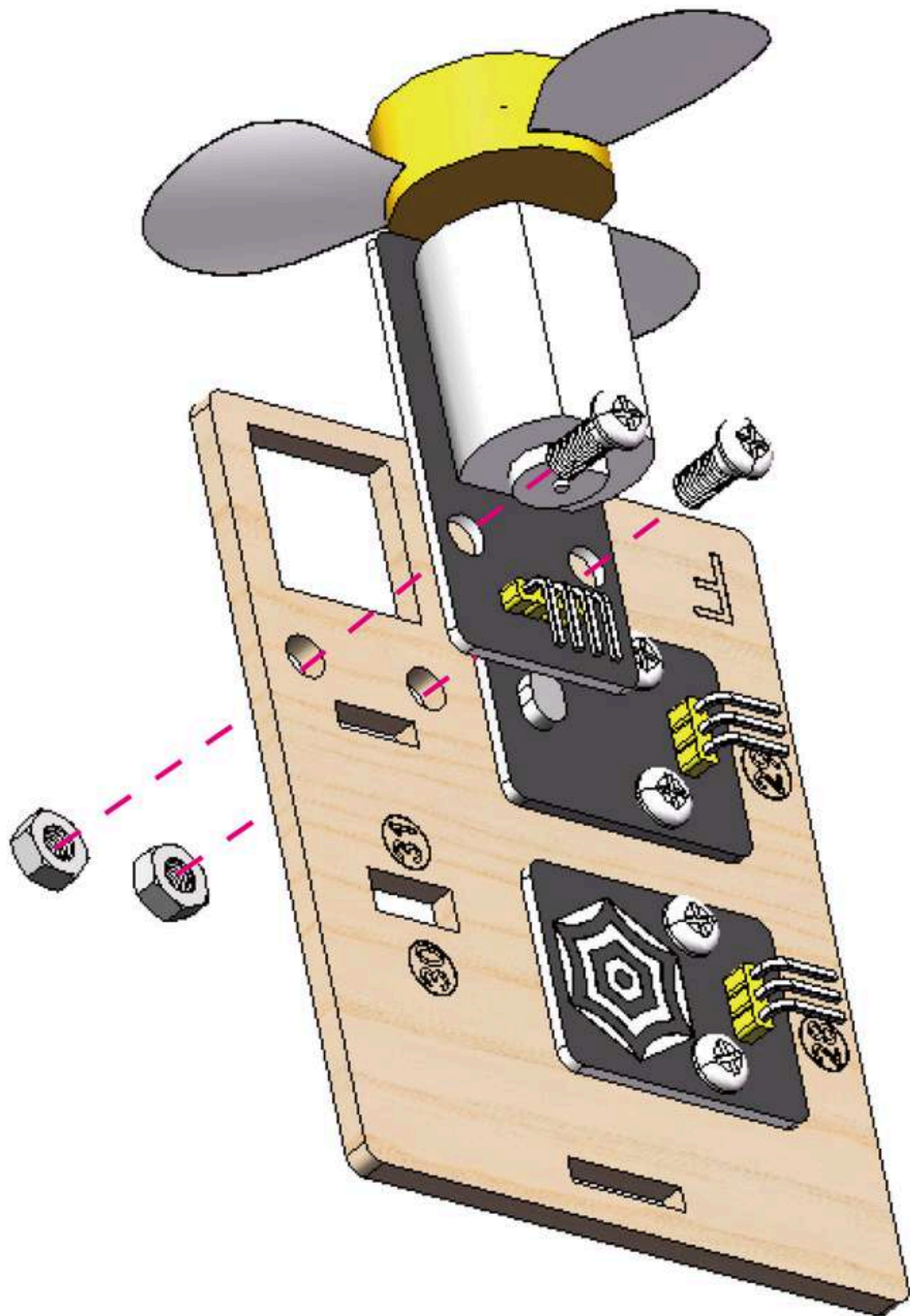
8.5



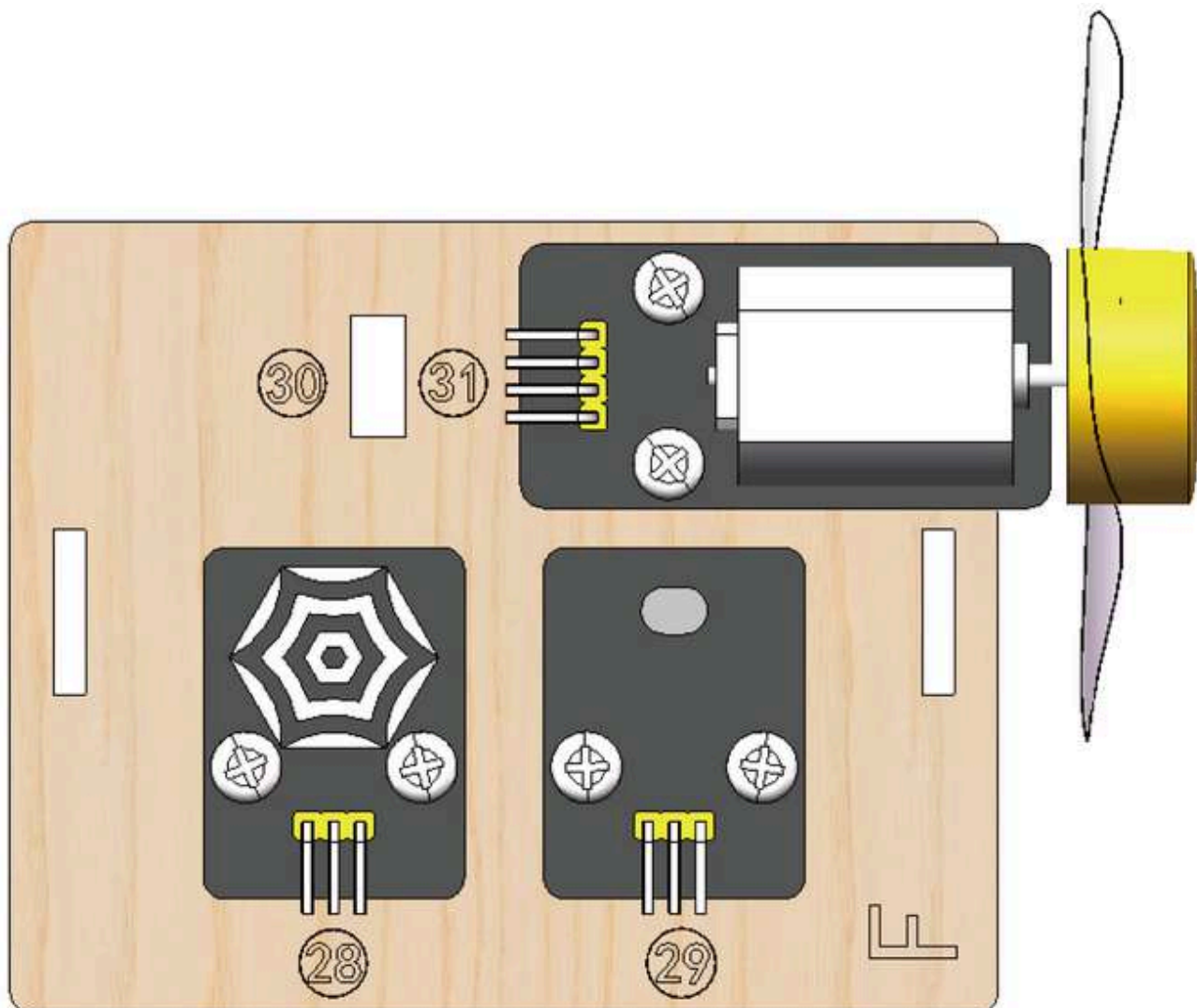
8.6



8.7



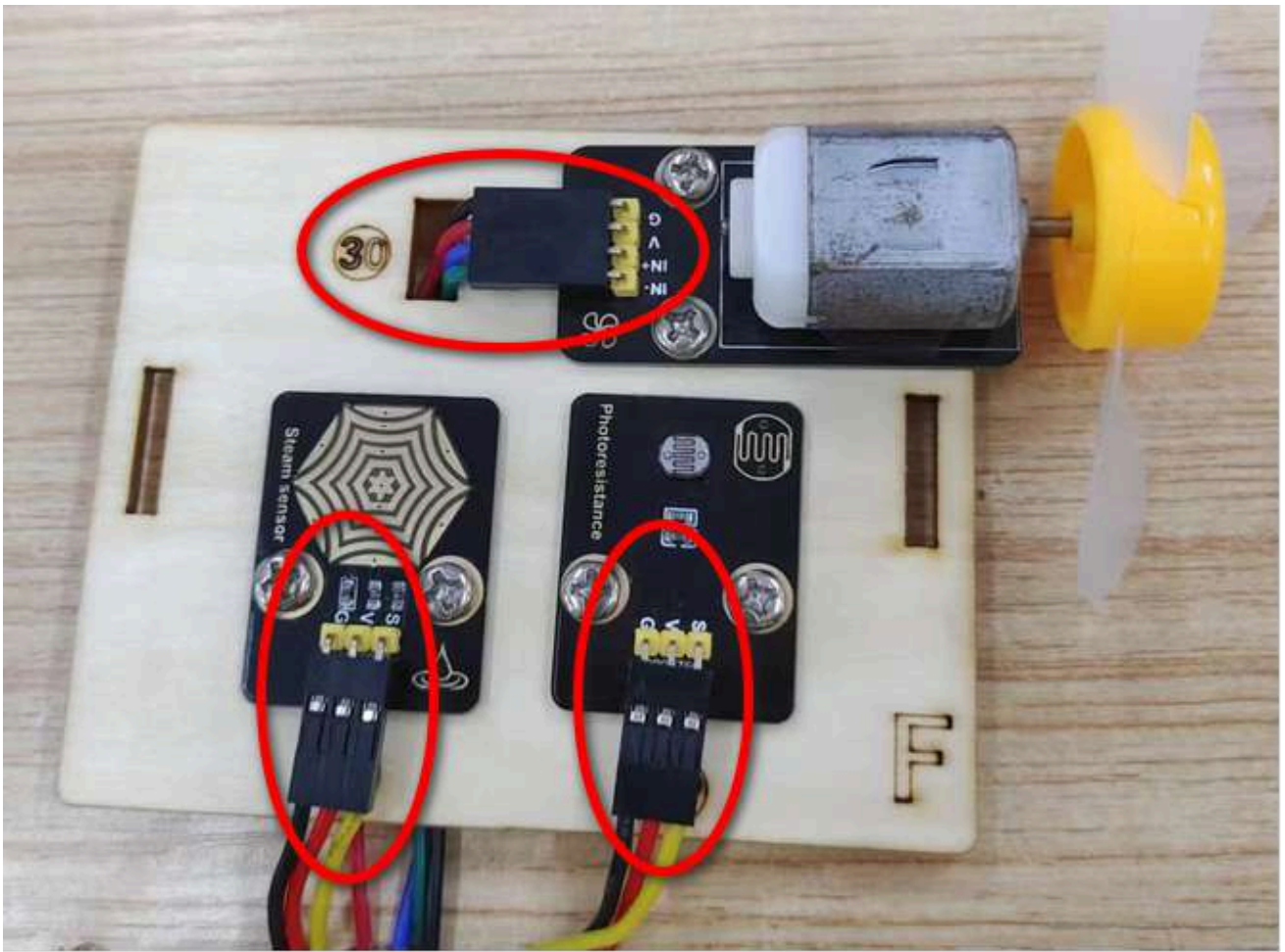
8.8 Prototip



8.9 Cablare

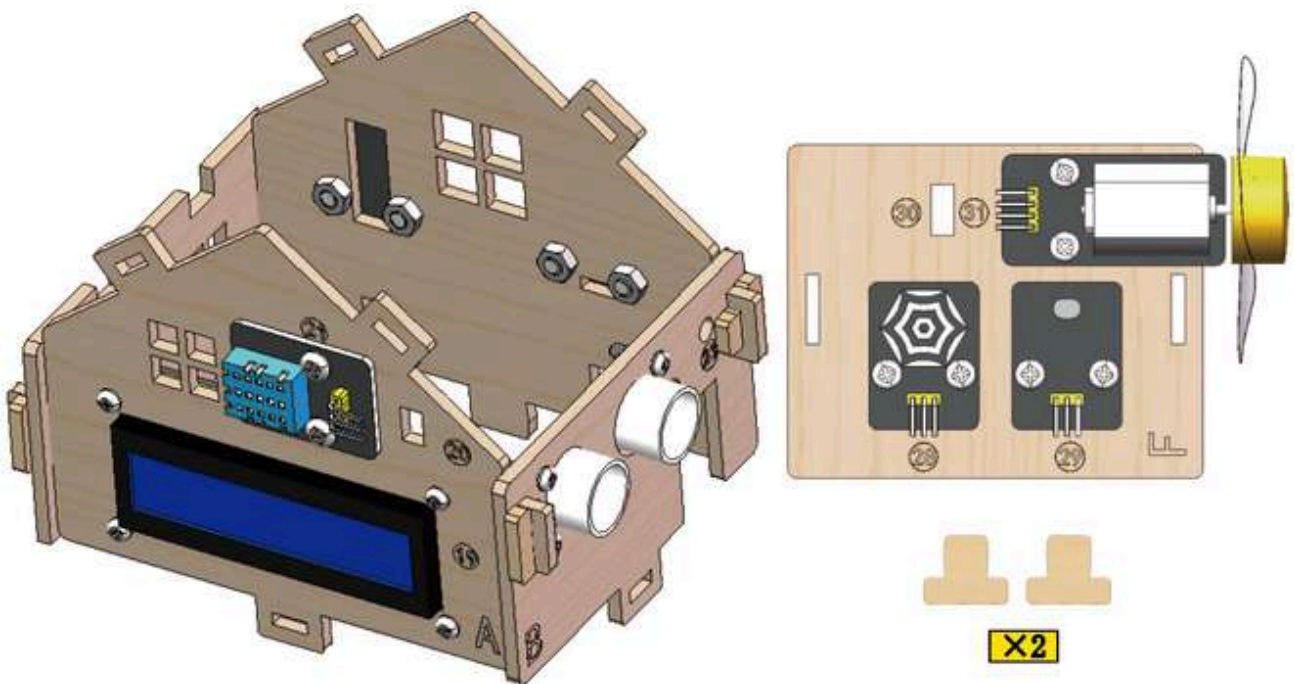
Modul	Sârmă
Evantai	4 pini (negru-roșu-albastru-verde)
Senzor de abur	3PIN 15cm
Fotorezistor	3PIN 15cm

Acordați atenție culorii firului Dupont: conectați galben la S, roșu la V, negru la G, albastru la IN+, verde la IN-.

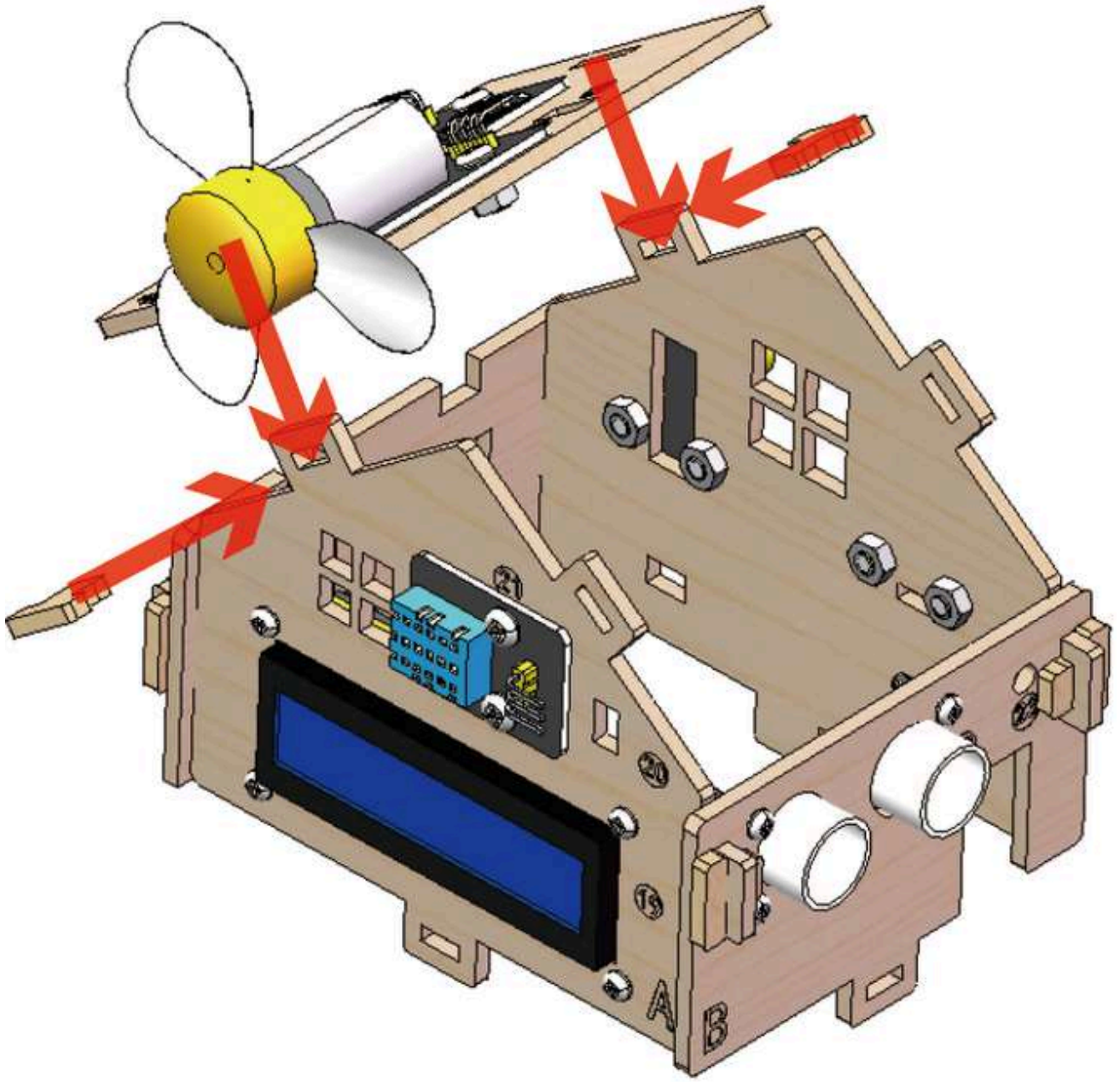


Pasul 9 Instalați acoperișul casei

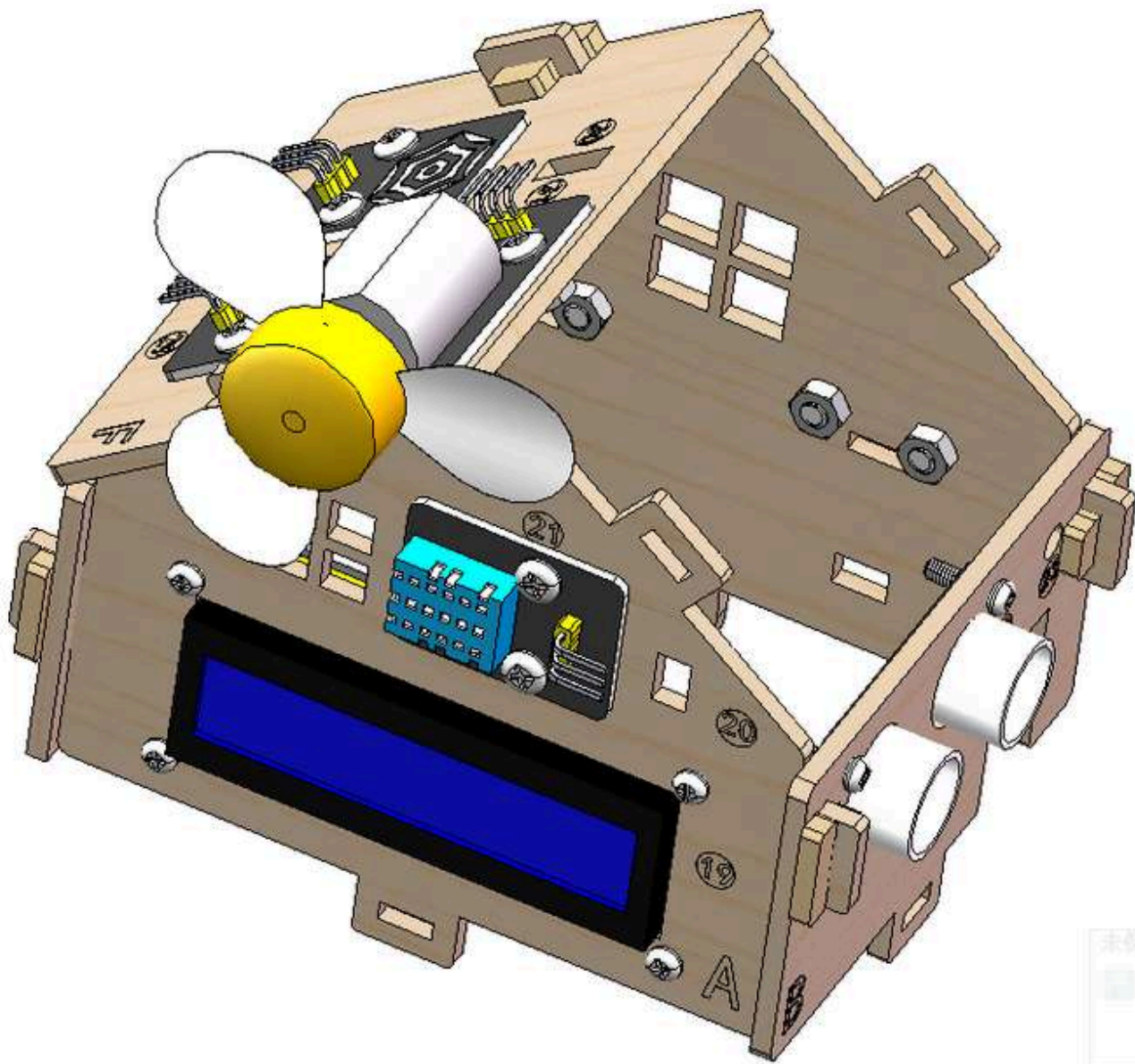
9.1 Componente necesare



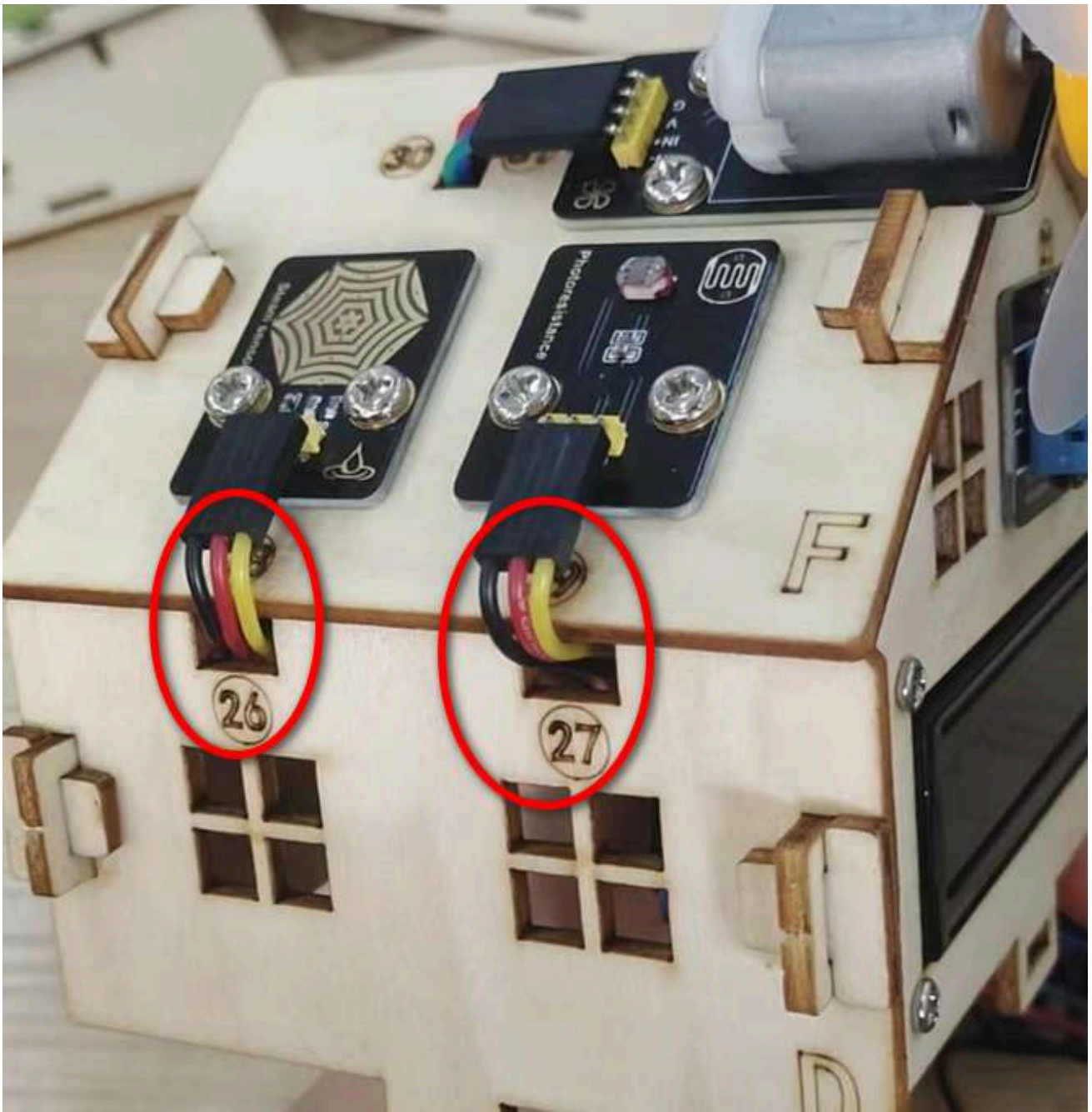
9.2



9.3

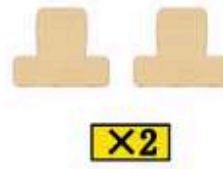
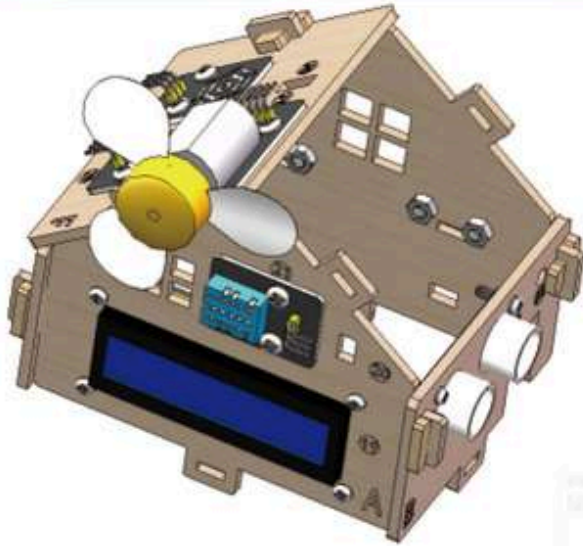
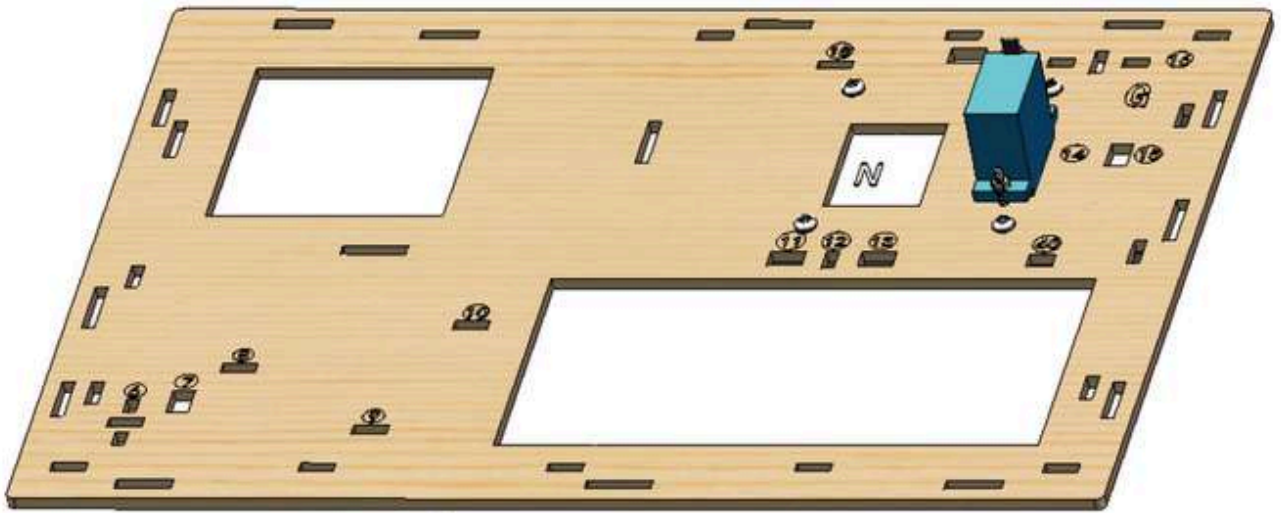


9.4 Păstrați firele organizate

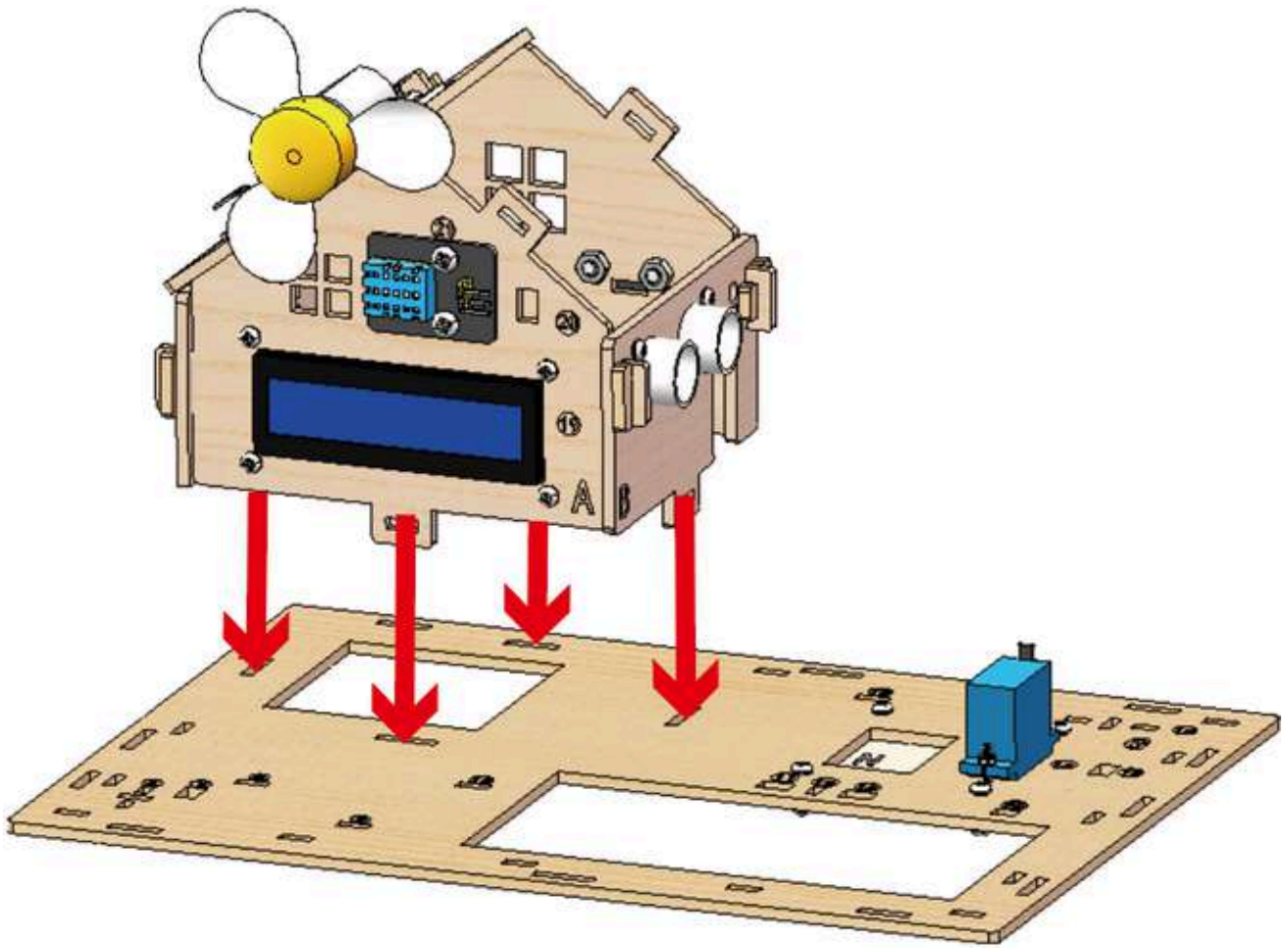


Pasul 10 Instalați casa și pământul

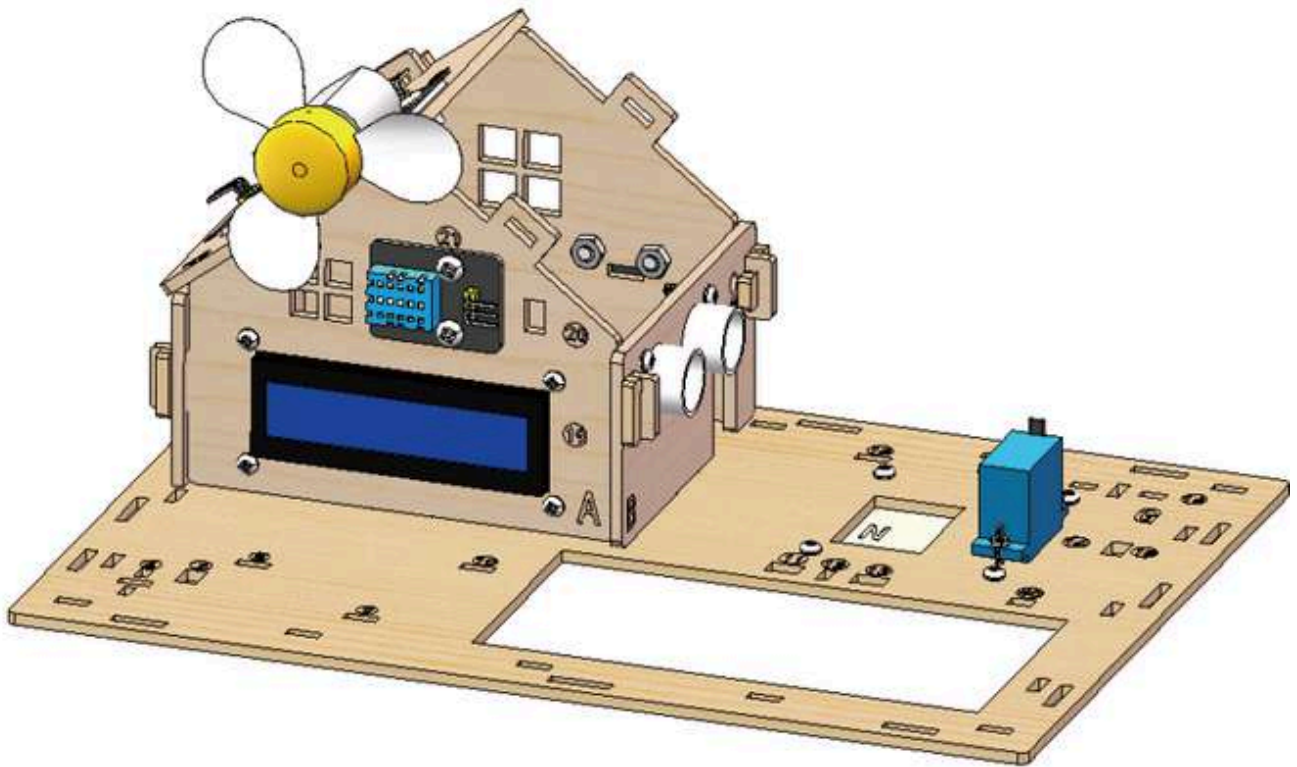
10.1 Componente necesare



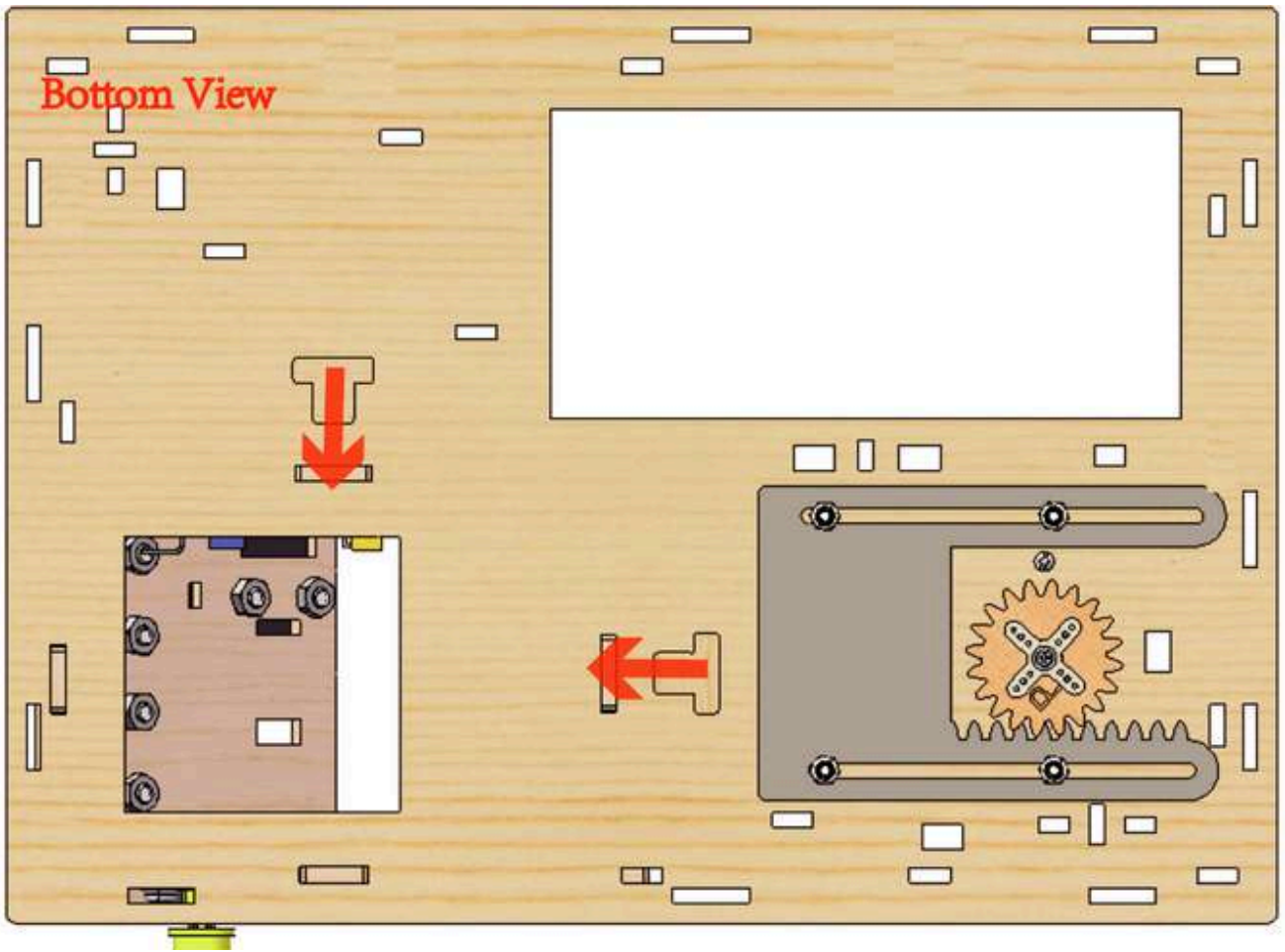
10.2



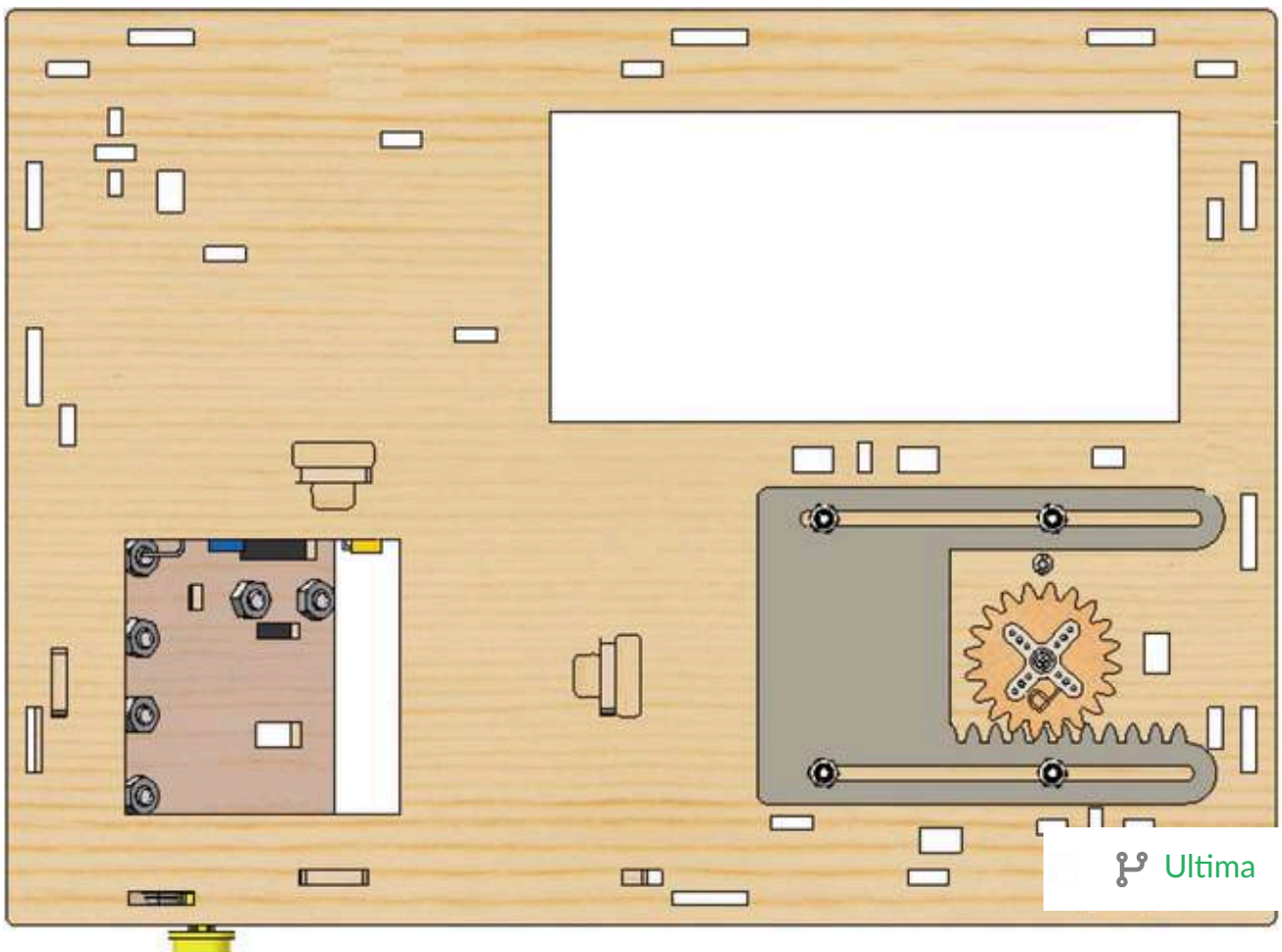
10.3



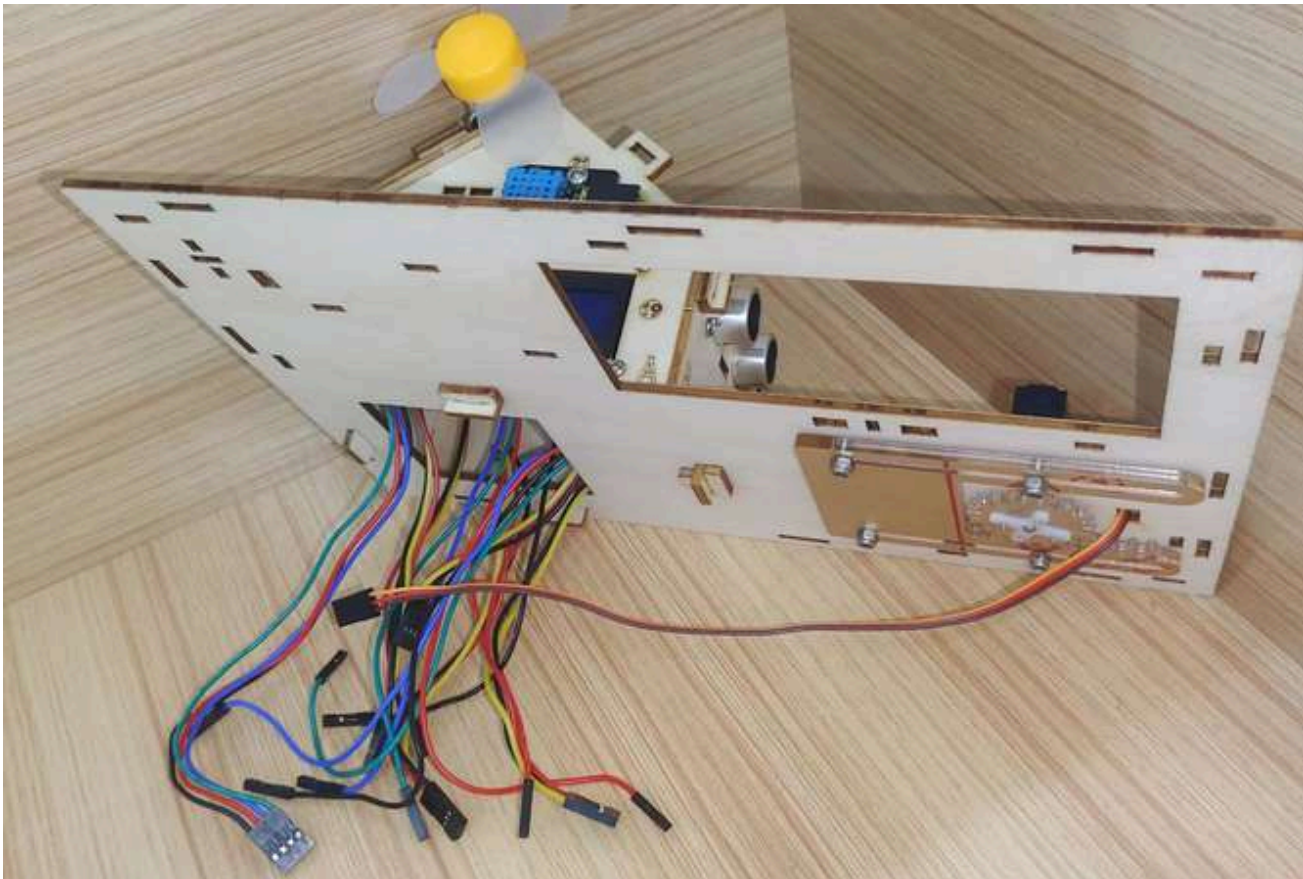
10.4 Bottom View



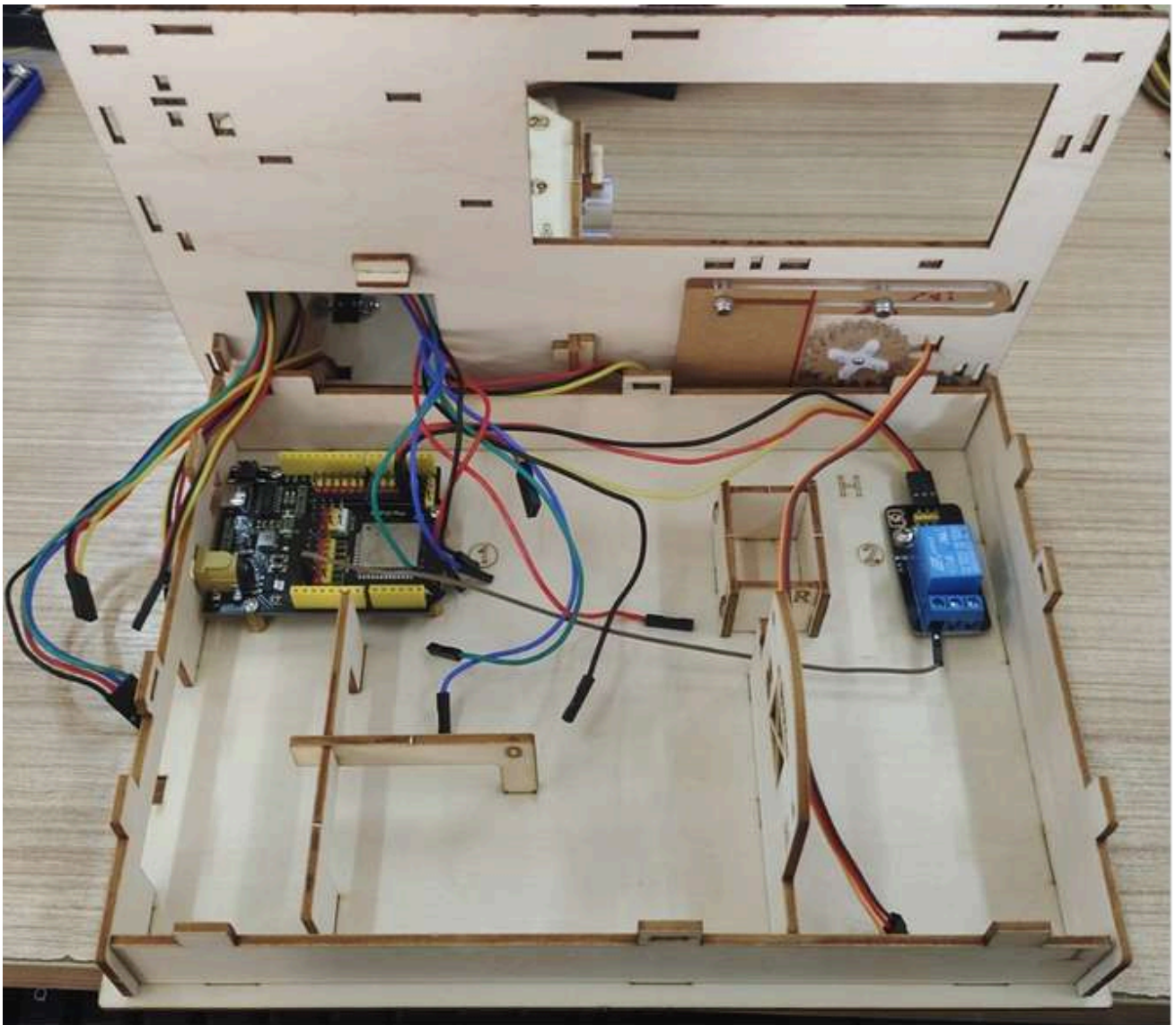
10.5



10.6 Arrange the wires



10.7



Pasul 11 Cablarea casei

11.1

Acordați atenție culorii firului Dupont: conectați galben la S, roșu la V, negru la G.

NU.	Componente	Fire	Știft
1	Evantai	4pin, împărțit Bla ck-Roșu-Albastru-Verde	io18
2	Senzor de mișcare PIR	3 pin 15cm	io23
3	Nasture	3 pin 15cm	io5
4	Modul cu ultrasunete	4 pin,Divizat Bla ck-Verde-Albastru-Roșu	D12
5	LCD 1602	4 pini, conectat	I2C
6	Temperatura și Senzor de umiditate	3 pini 20cm	
7	Senzor de abur	3 pin 15cm	io35

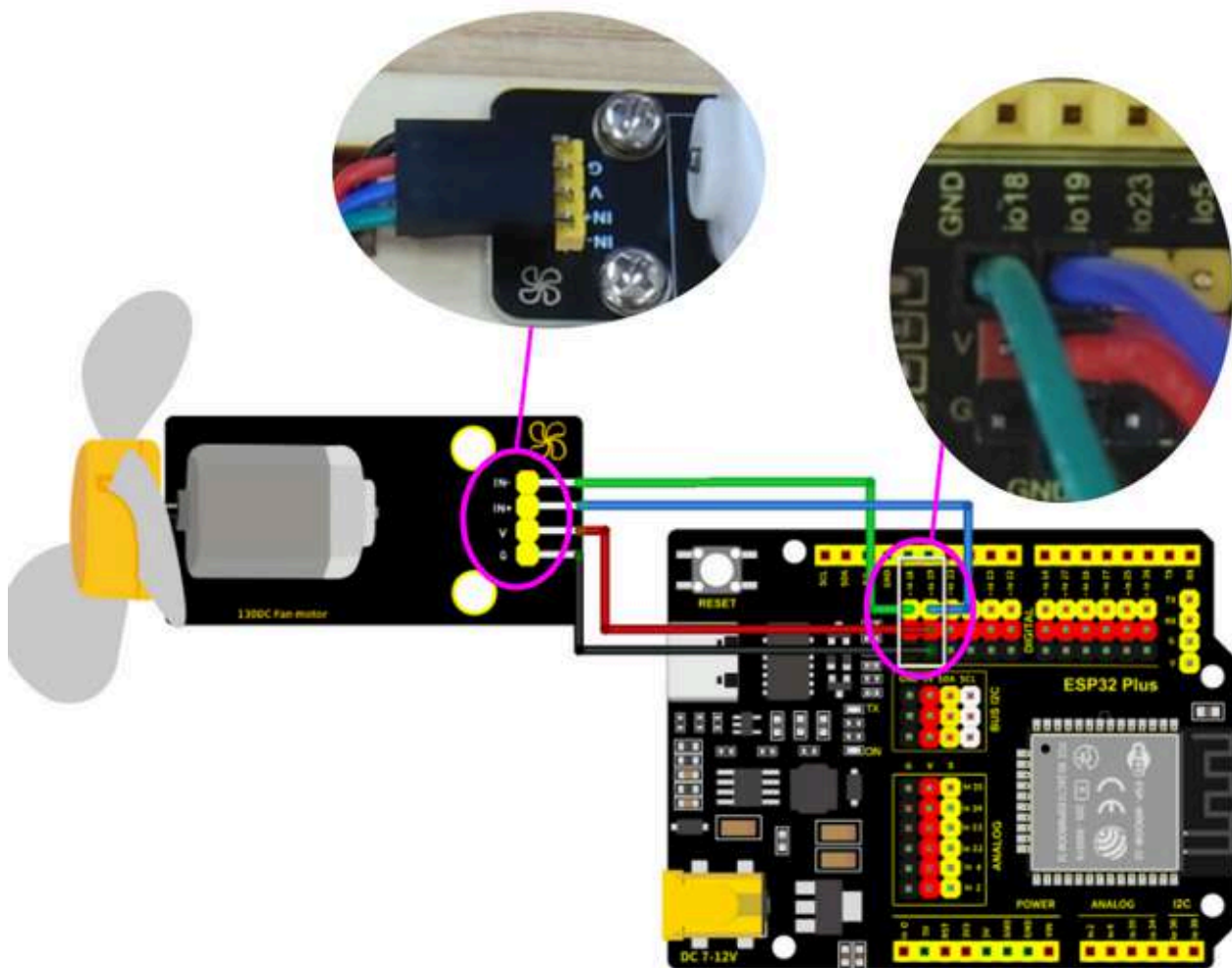
NU.	Componente	Fire	Știft
8	Fotorezistor	3 pin 15cm	io34
9	Servo	-	io26
10	Sonerie	3 pini 20cm	io16
11	LED	3 pini 20cm	io27
12	Nivelul apei Senzor	3pin 25cm	io33
13	Umiditatea solului Senzor	3 pini 20cm	io32
14	Pompă de apă	3 pini 20cm	io25

11.2 Ventilator

Treceți firul Dupont conectat la ventilator prin orificiul **marcat 30** de pe placa de lemn.

Componente	Sârmă	Știfturi de placă ESP32
Evantai	4 pini împărțiți * (negru-roșu-albastru-verde)*	io18(IN-), io19(IN+)

Pin modul	Culoarea sârmei
ÎN-	VERDE
IN+	ALBASTRU
V	ROȘU
G	NEGRU



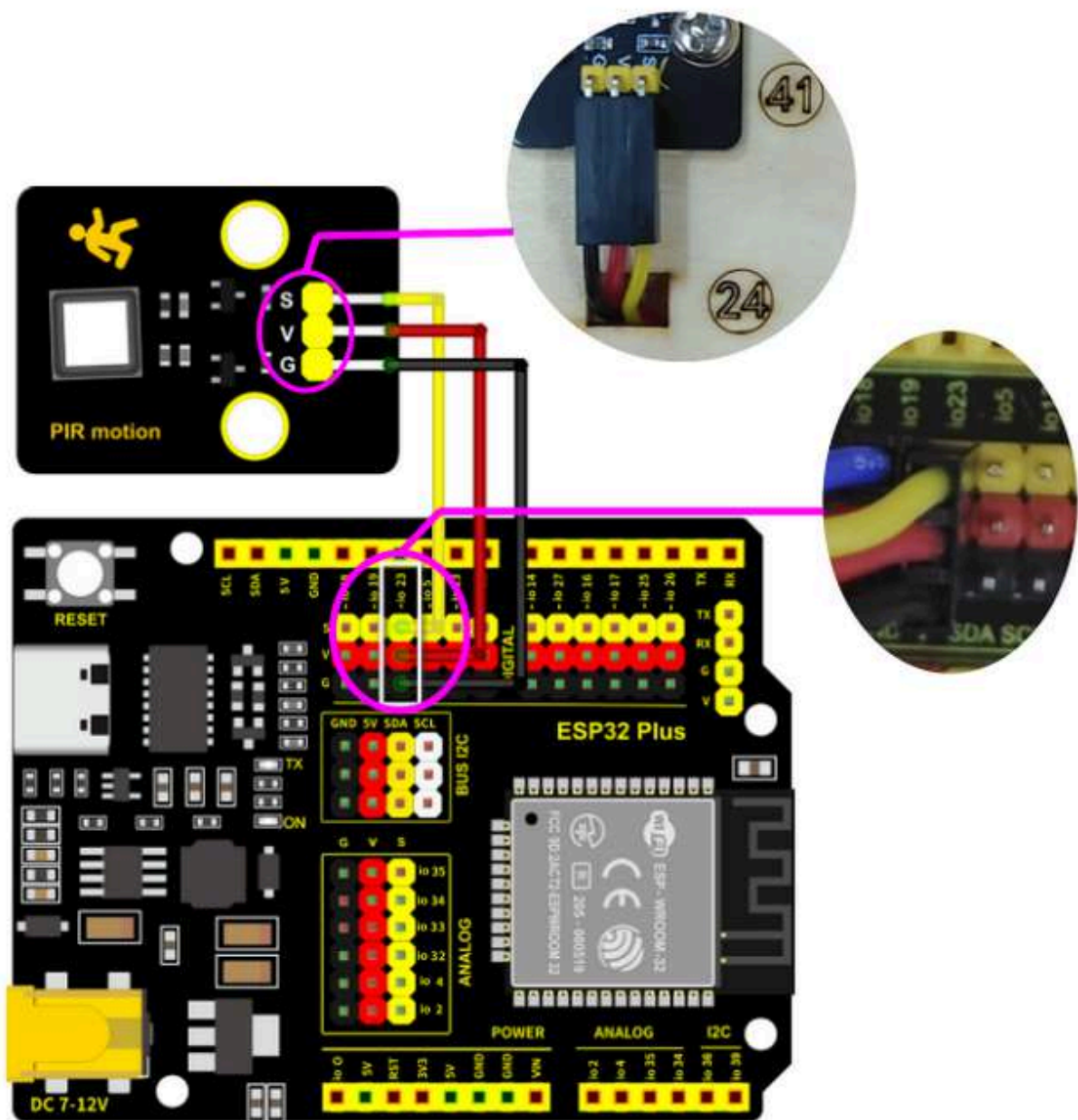
11.3 Senzor de mișcare PIR

Treceți firul Dupont conectat la senzorul de mișcare PIR prin orificiu marcat 24 pe placa de lemn.

Componentă	Sârmă	Pin placă ESP32
Senzor de mișcare PIR	3PIN 15cm	io23

Conectați roșu la V, negru la G, galben la S.

Pin modul	Culoarea sârmei	Pin placă ESP32
V	ROȘU	V
G	NEGRU	G
S	GALBEN	io23



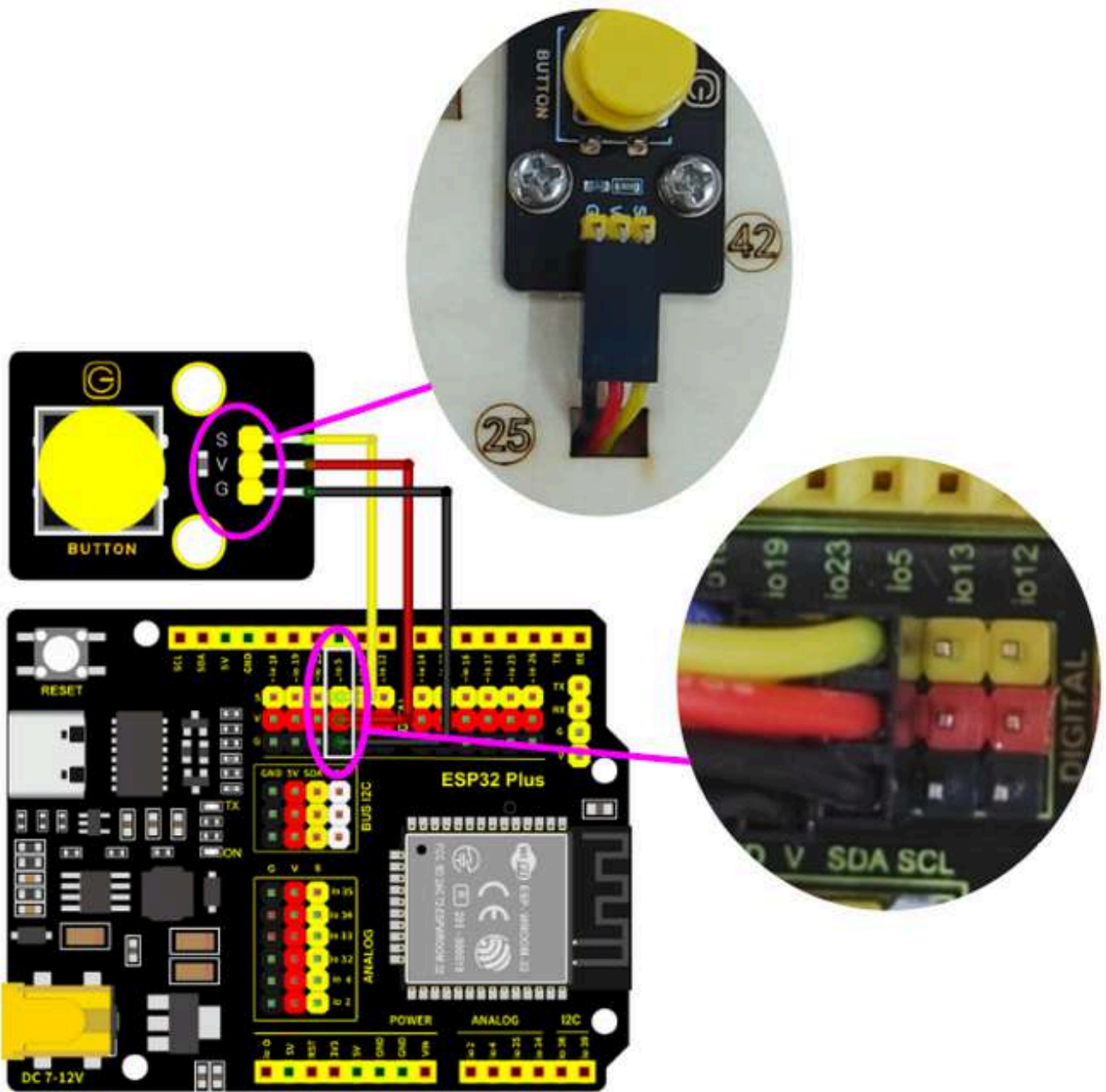
11.4 Modul buton

Treceți firul Dupont conectat la modulul buton prin orificiu marcat cu 25 pe placa de lemn.

Componentă	Sârmă	Pin placă ESP32
Nasture	3PIN 15cm	io5

Conectați roșu la V, negru la G, galben la S.

Pin modul	Culoarea sârmei	Pin placă ESP32
V	ROȘU	V
G	NEGRU	G
S	GALBEN	io5

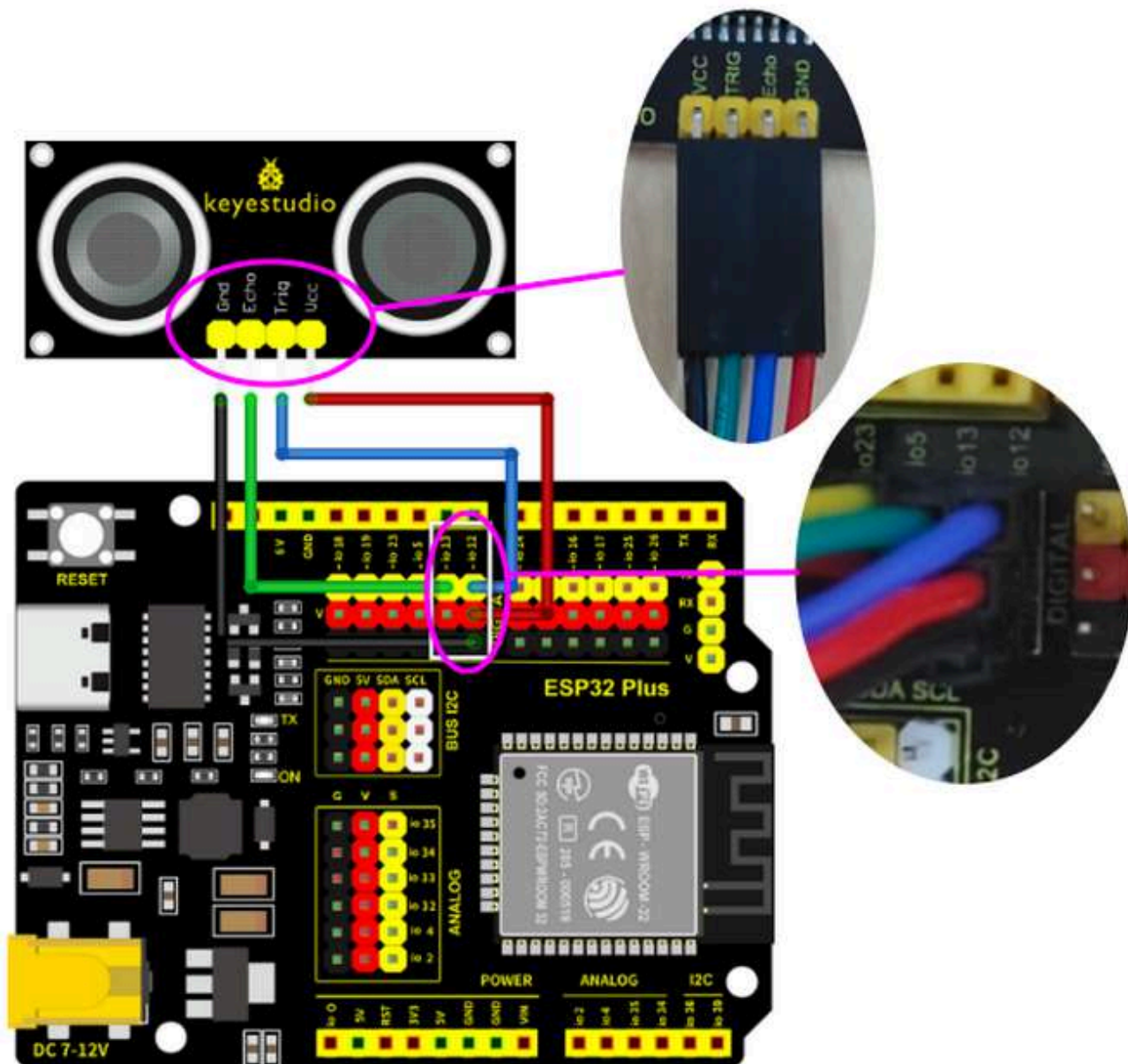


11.5 Modul cu ultrasunete

Componentă	Sârmă	Știfturi de placă ESP32
Modul cu ultrasunete	4PIN divizat (Negru-Verde-Albastru-Roșu)	io13(ECHO), io12(TRIG)

Conectați roșu la V, negru la G, albastru la io12, verde la io13.

Pin modul	Culoarea sârmei	Pin placă ESP32
V	ROȘU	V (io12)
G	NEGRU	G (io12)
ECOU	VERDE	io13
TRIG	ALBASTRU	io12

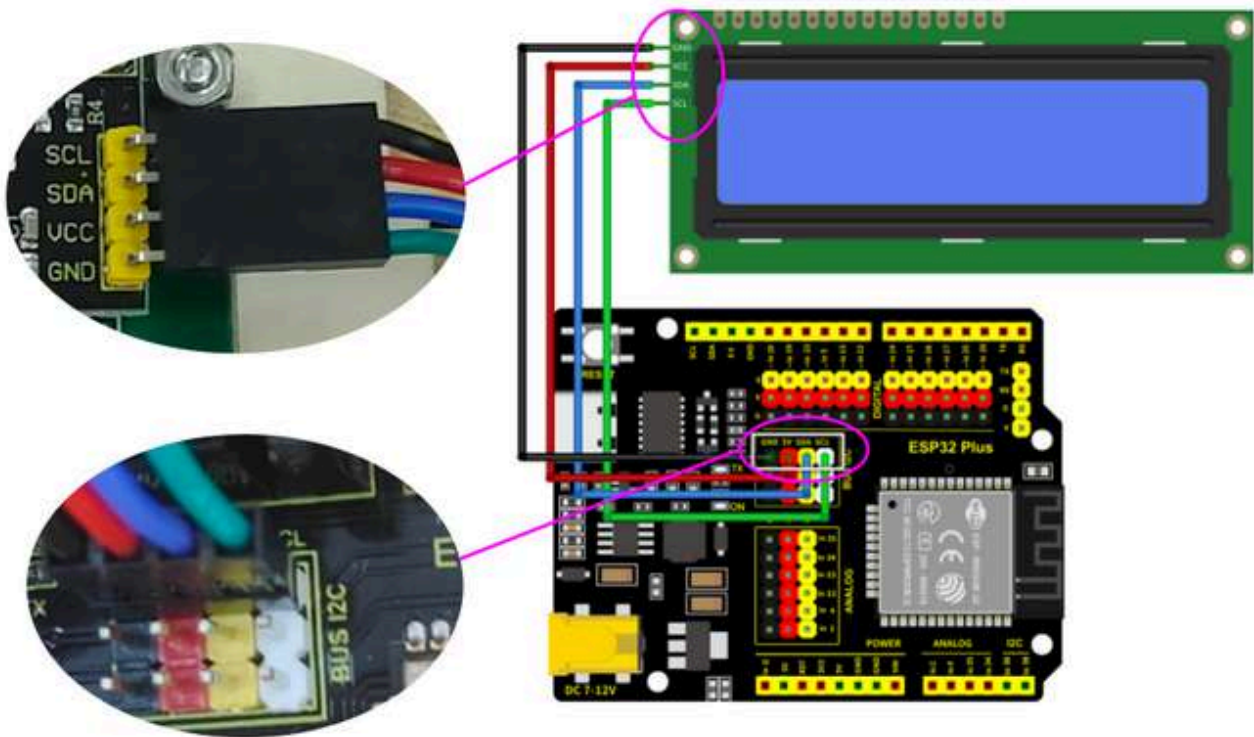


11.6 LCD 1602

Componentă	Sârmă	Știfturi de placă ESP32
LCD1602	Conectat cu 4 pini (negru-roșu-albastru-verde)	I2C

Conectați roșu la V, negru la G, albastru la SDA, verde la SCL.

Pin modul	Culoarea sârmei	Pin placă ESP32
V	ROȘU	V
G	NEGRU	G
SCL	VERDE	SCL
SDA	ALBASTRU	SDA



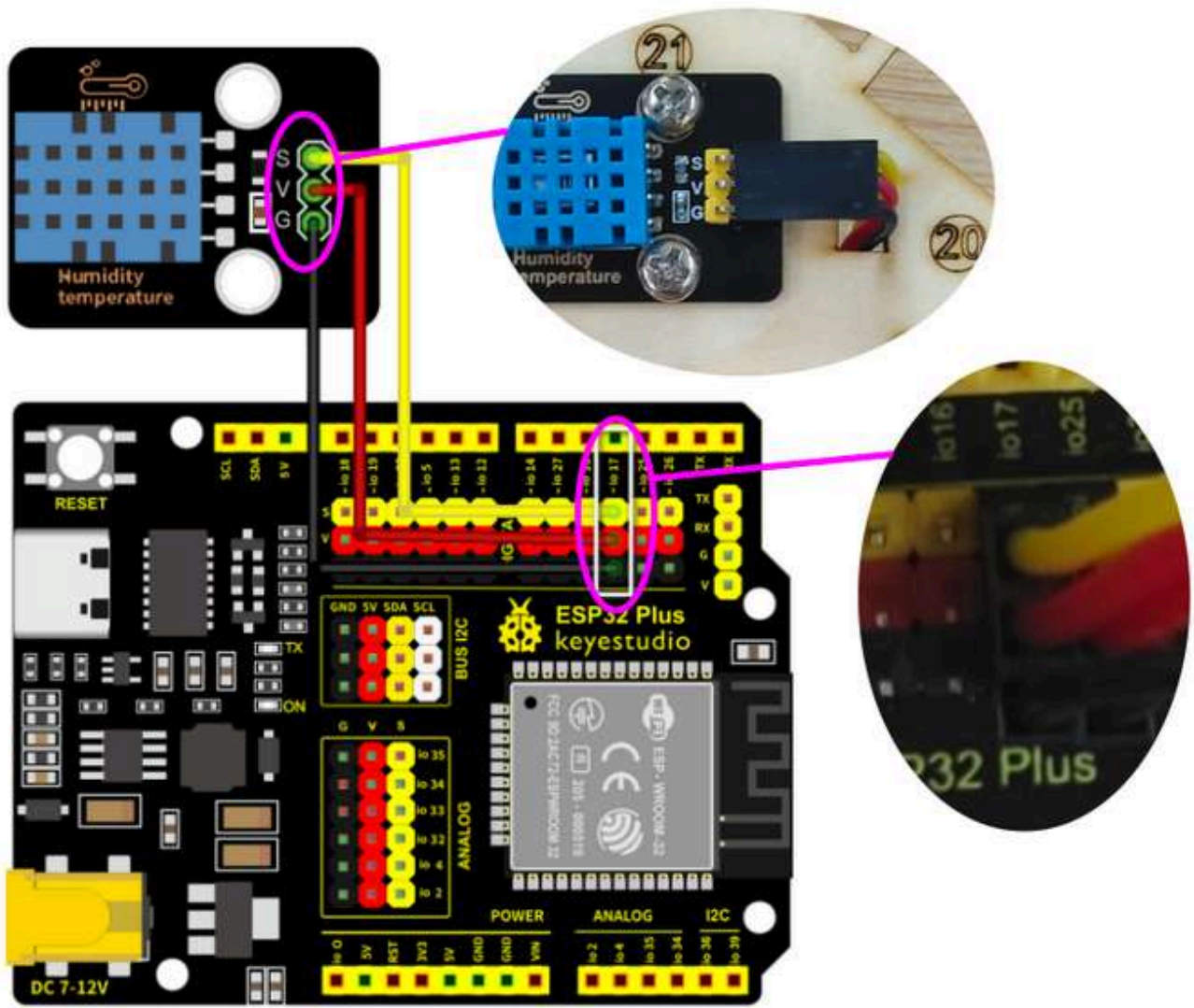
11.7 Senzor de temperatură și umiditate

Treceți firul Dupont conectat la modulul buton prin orificiu marcat 20 pe placa de lemn.

Componentă	Sârmă	Știfturi de placă ESP32
Senzor de temperatură și umiditate	3PIN 20cm	io17

Conectați roșu la V, negru la G, galben la io17.

Pin modul	Culoarea sârmei	Pin placă ESP32
V	ROȘU	V
G	NEGRU	G
S	GALBEN	io17

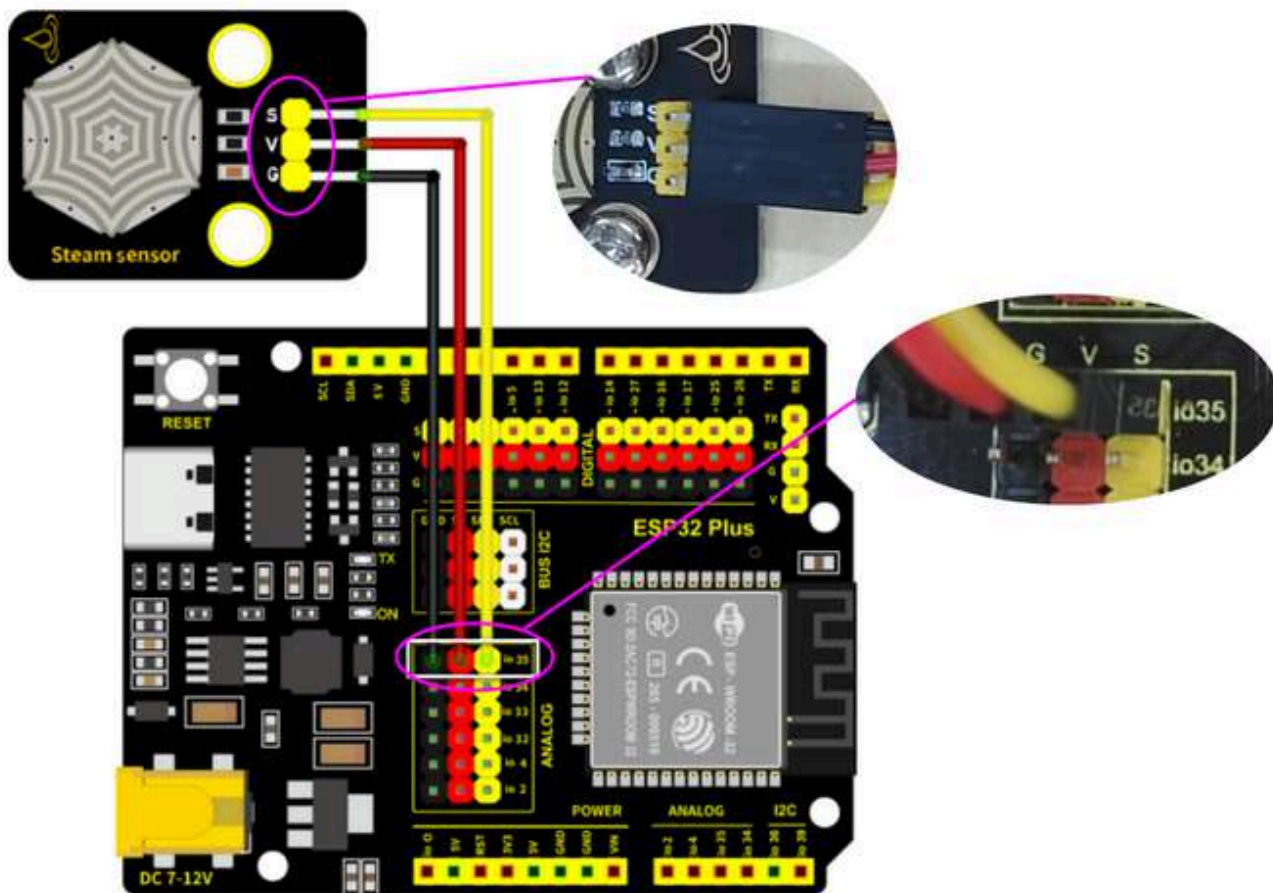


11.8 Senzor de abur

Componentă	Sârmă	Pin placă ESP32
Senzor de abur	3PIN 15cm	io35

Conectați roșu la V, negru la G, galben la io35.

Pin modul	Culoarea sârmei	Pin placă ESP32
V	ROȘU	V
G	NEGRU	G
S	GALBEN	io35

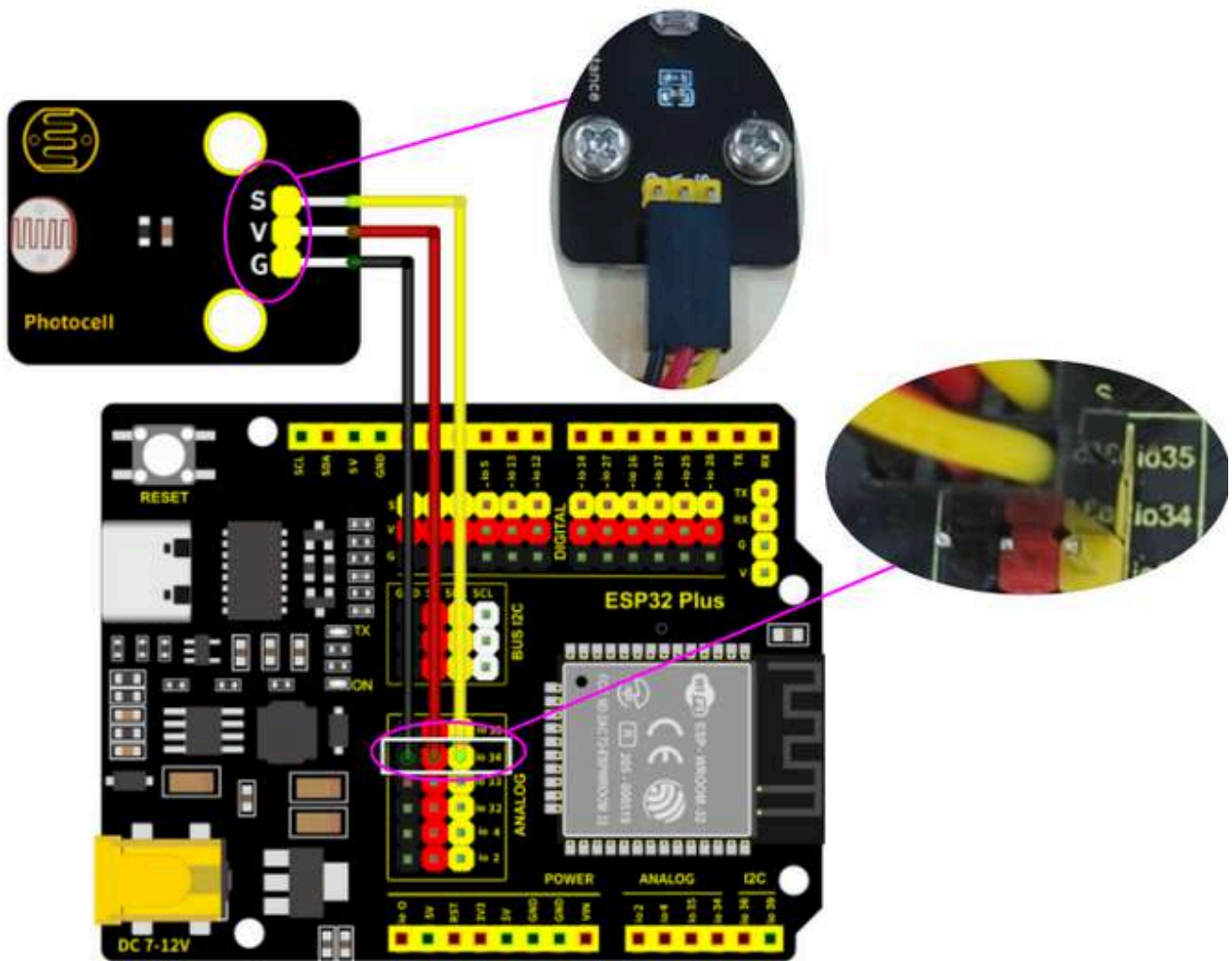


11.9 Fotorezistență

Componentă	Sârmă	Pin placă ESP32
Fotorezistor	3PIN 15cm	io34

Conectați roșu la V, negru la G, galben la io34.

Pin modul	Culoarea sârmei	Pin placă ESP32
V	ROȘU	V
G	NEGRU	G
S	GALBEN	io34



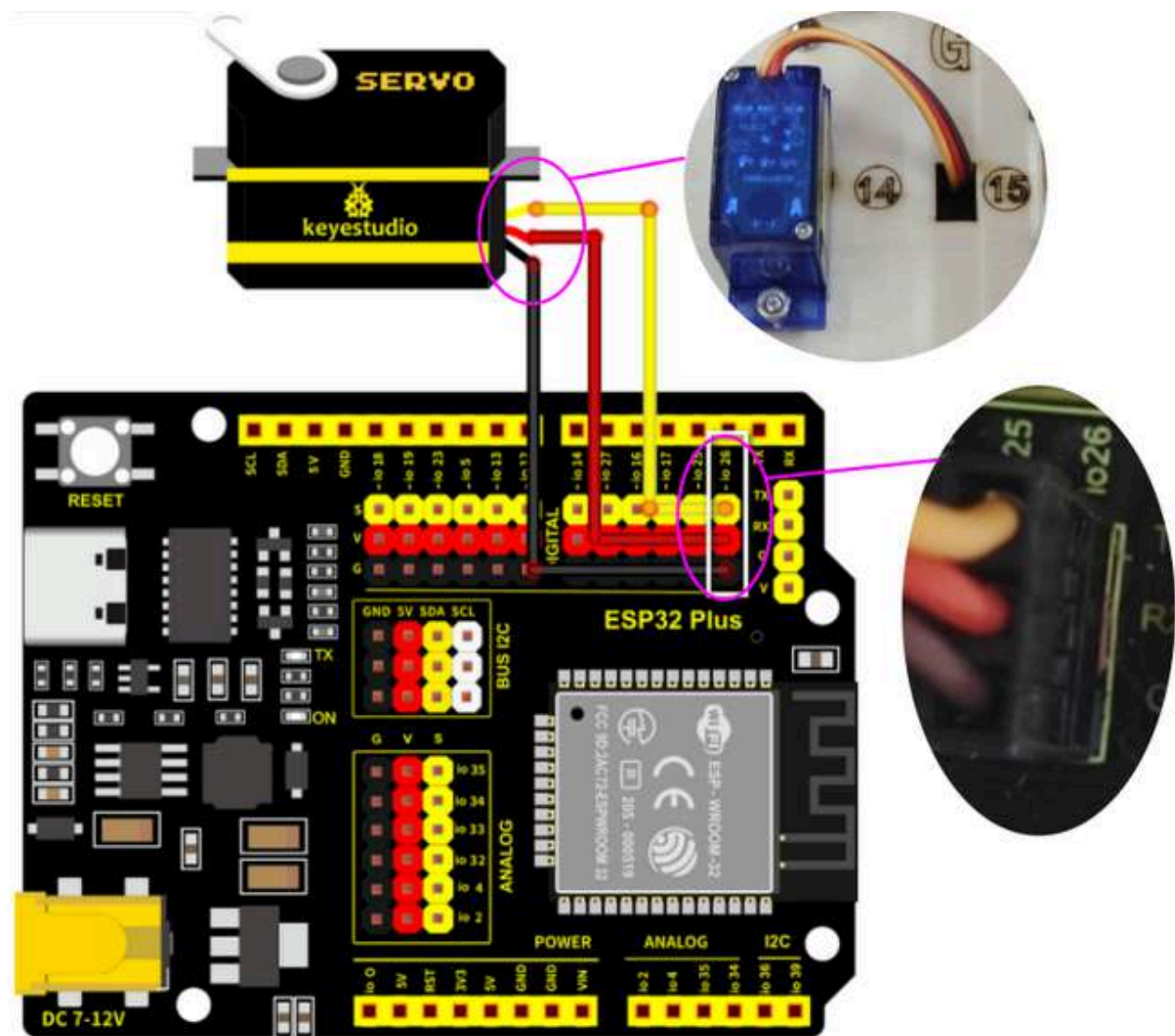
11.10 Servo

Treceți firul Servo prin orificiul 15, apoi conectați-l la Placă ESP32.

Componentă	Sârmă	Pin placă ESP32
Servo	3 pini	io26

Conectați roșu la V, negru la G, galben la io26.

Pin de placă	Culoarea sârmei
V	ROȘU
G	NEGRU
IO26	GALBEN



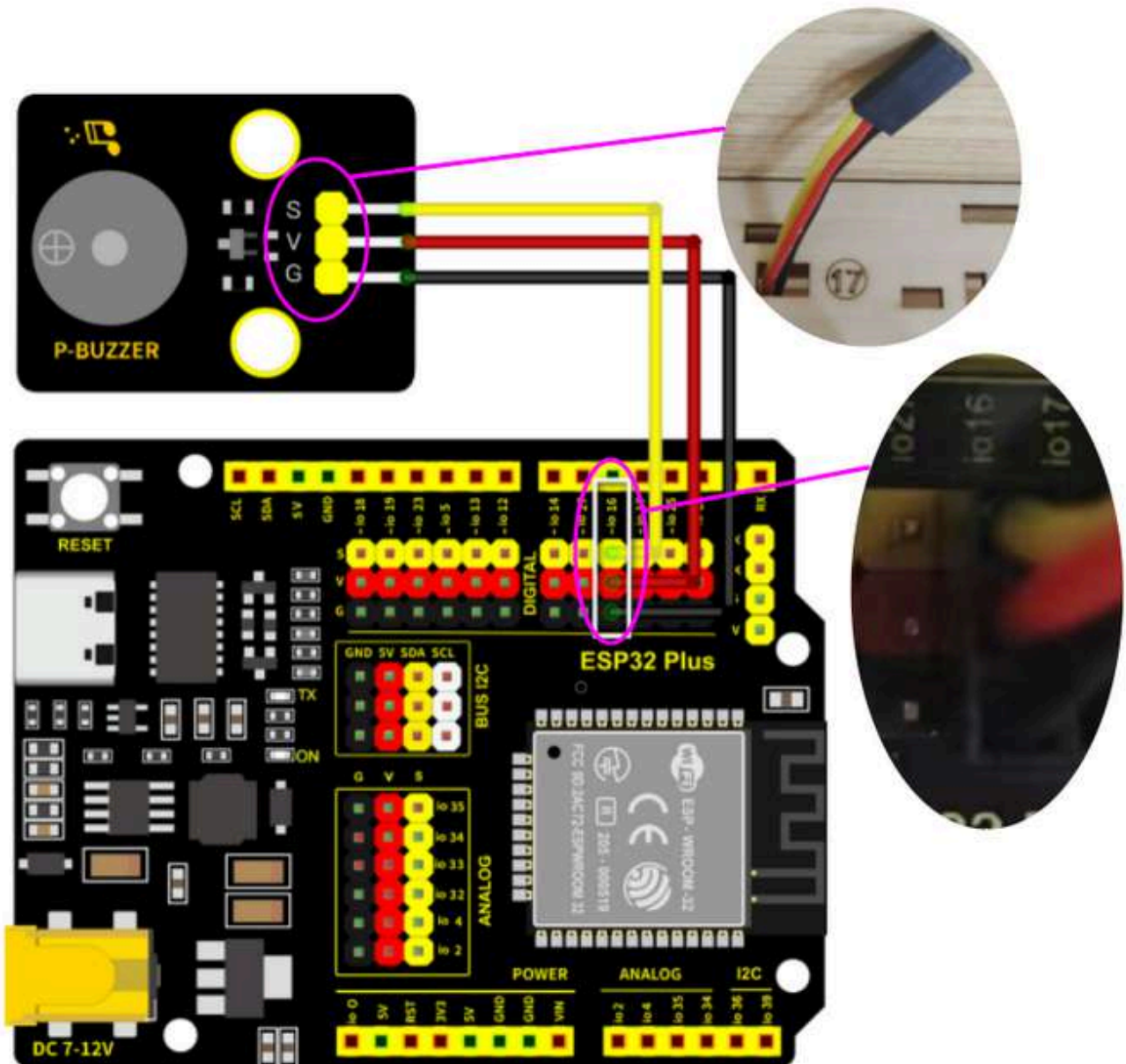
11.11 Sonerie

Treceți firul soneriei prin orificiul 17, apoi conectați-l la Placă ESP32.

Componentă	Sârmă	Pin placă ESP32
Sonerie	3PIN 20cm	io16

Conectați roșu la V, negru la G, galben la io16.

Pin modul	Culoarea sârmei	Pin placă ESP32
V	ROȘU	V
G	NEGRU	G
S	GALBEN	io16



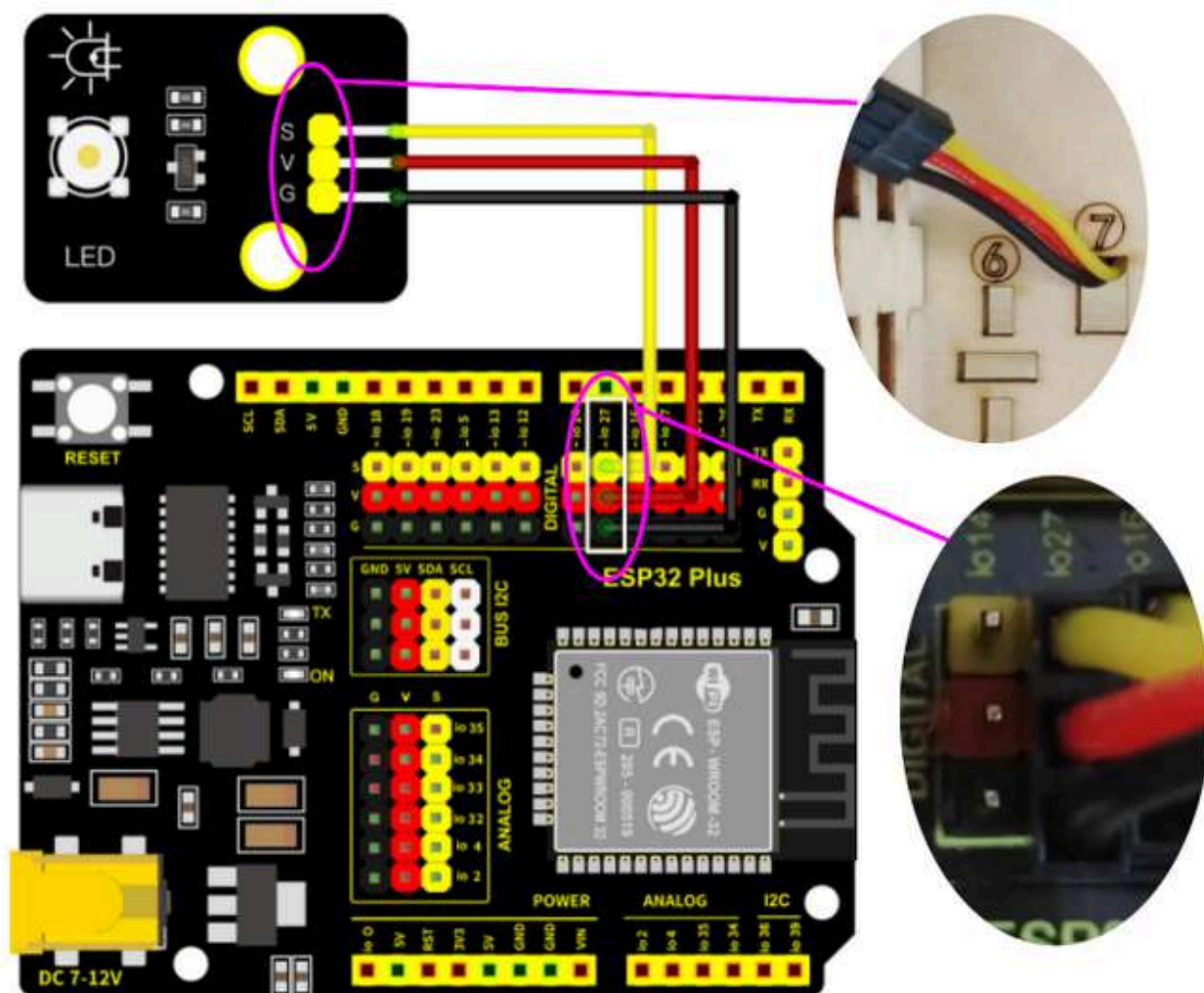
11.12 LED

Pass the wire of LED through the Hole 7, and then connect it to ESP32 board.

Component	Wire	ESP32 Board Pin
LED	3PIN 20cm	io27

Connect red to V, black to G, yellow to io27.

Module Pin	Wire Color	ESP32 Board Pin
V	ROȘU	V
G	NEGRU	G
S	GALBEN	io27



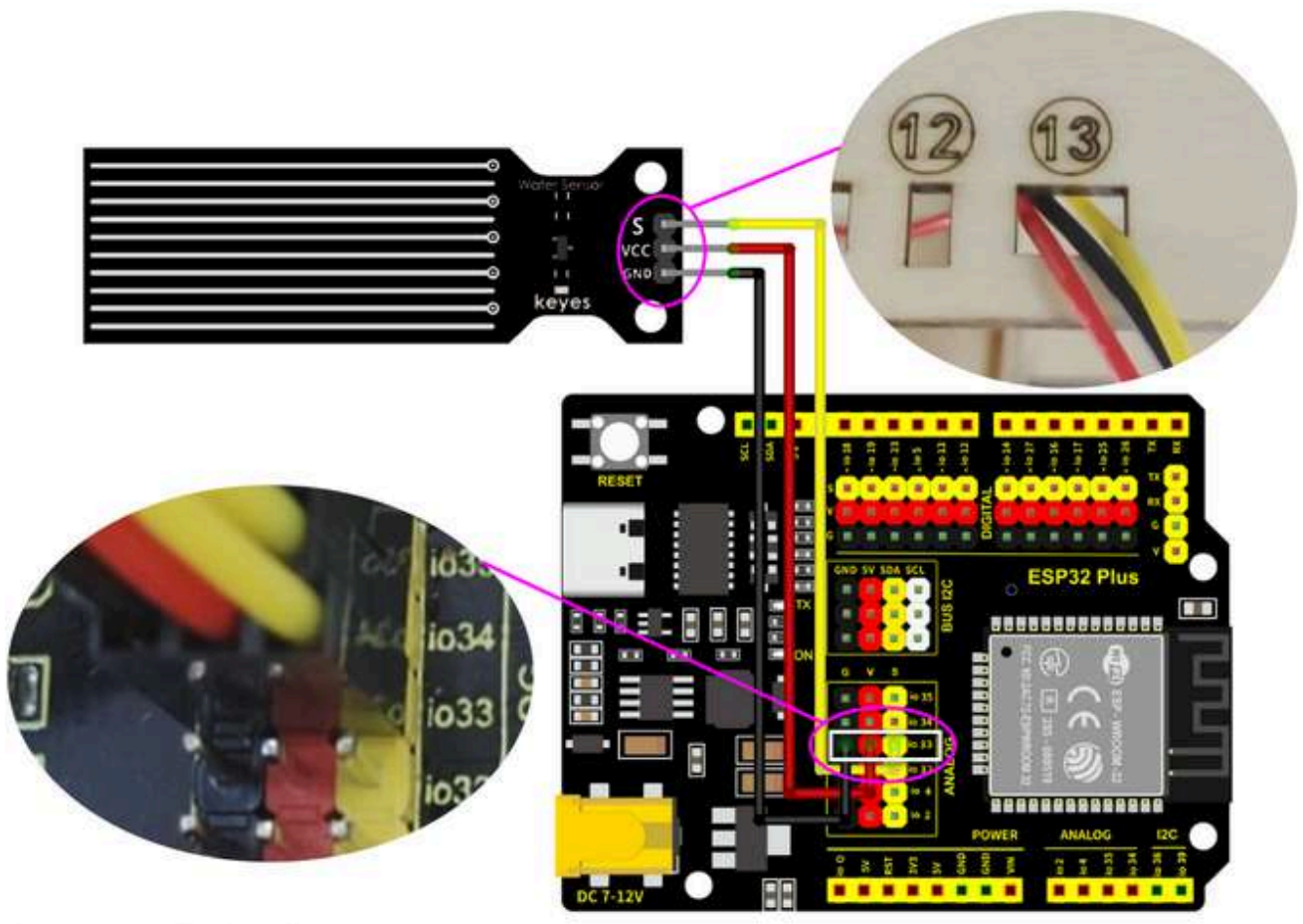
11.13 Senzor de pârghie de apă

Treceți firul senzorului de nivel al apei prin orificiul 13, apoi conectați-l la placa ESP32.

Componentă	Sârmă	Pin placă ESP32
Senzor de pârghie de apă	3PIN 25cm	io33

Conectați roșu la V, negru la G, galben la io33.

Pin modul	Culoarea sârmei	Pin placă ESP32
V	ROȘU	V
G	NEGRU	G
S	GALBEN	io33



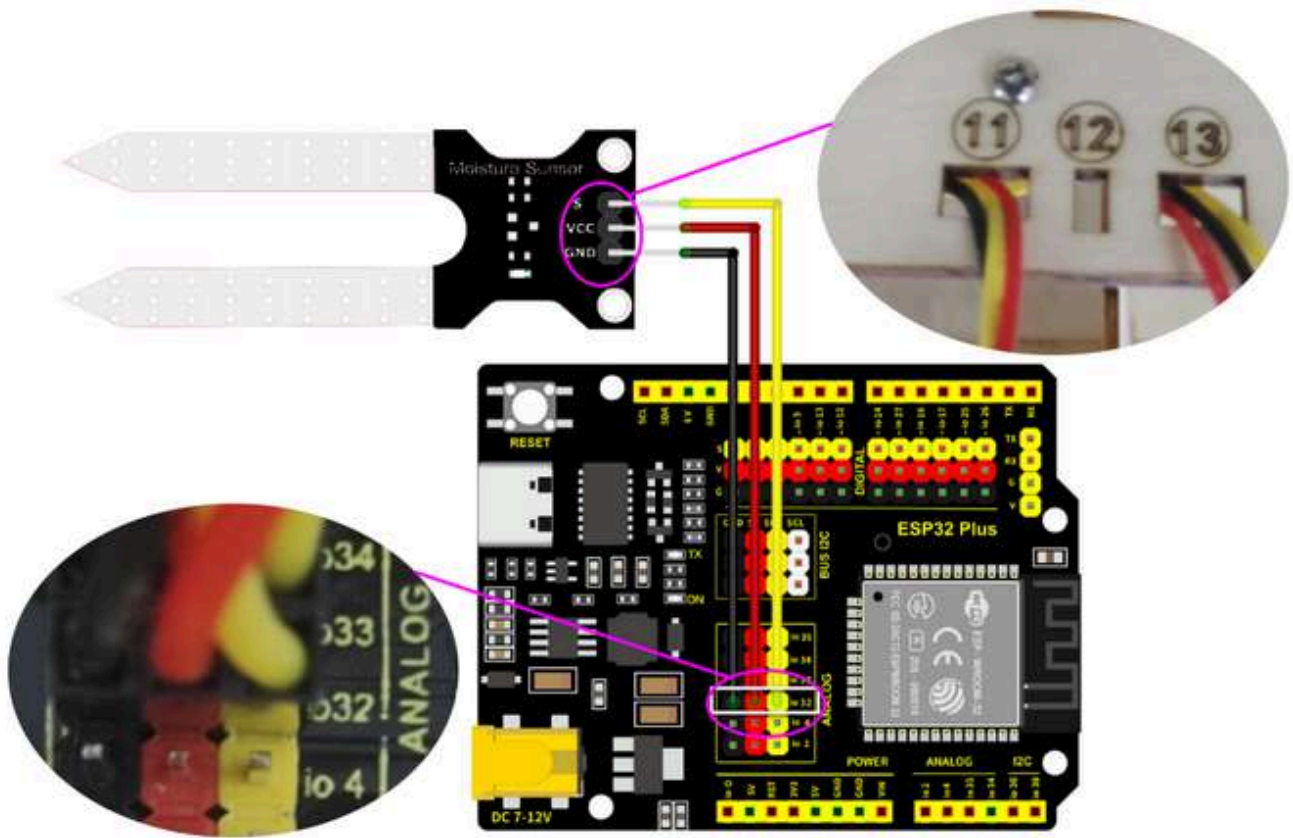
11.14 Senzor de umiditate a solului

Treceți firul senzorului de umiditate a solului prin orificiul 11, apoi conectați-l la placa ESP32.

Componentă	Sârmă	Pin placă ESP32
Senzor de umiditate a solului	3PIN 20cm	io32

Conectați roșu la V, negru la G, galben la io32.

Pin modul	Culoarea sârmei	Pin placă ESP32
V	ROȘU	V
G	NEGRU	G
S	GALBEN	io32

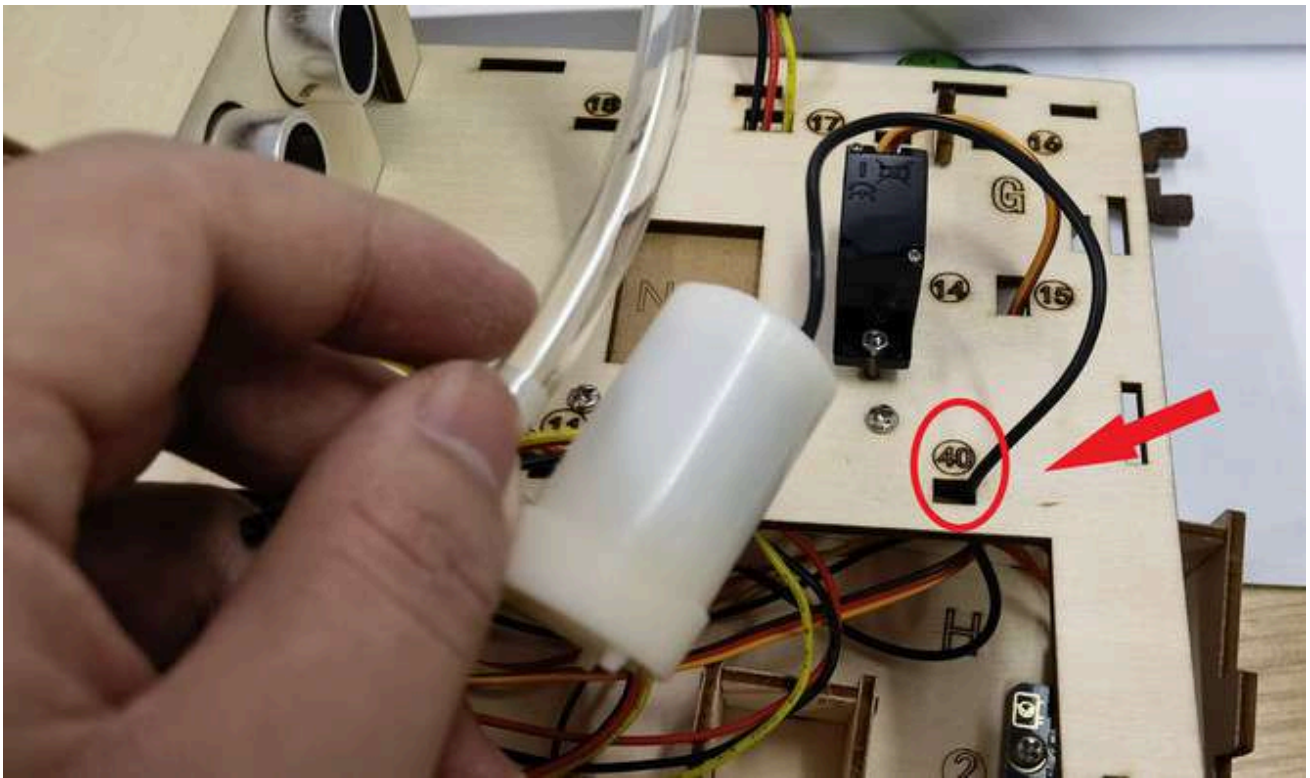


11.15 Modul releu

Componentă	Sârmă	Pin placă ESP32
Modul releu	3 pini	io25

Pin modul	Culoarea sârmei	Pin placă ESP32
V	ROȘU	V
G	NEGRU	G
S	GALBEN	io25

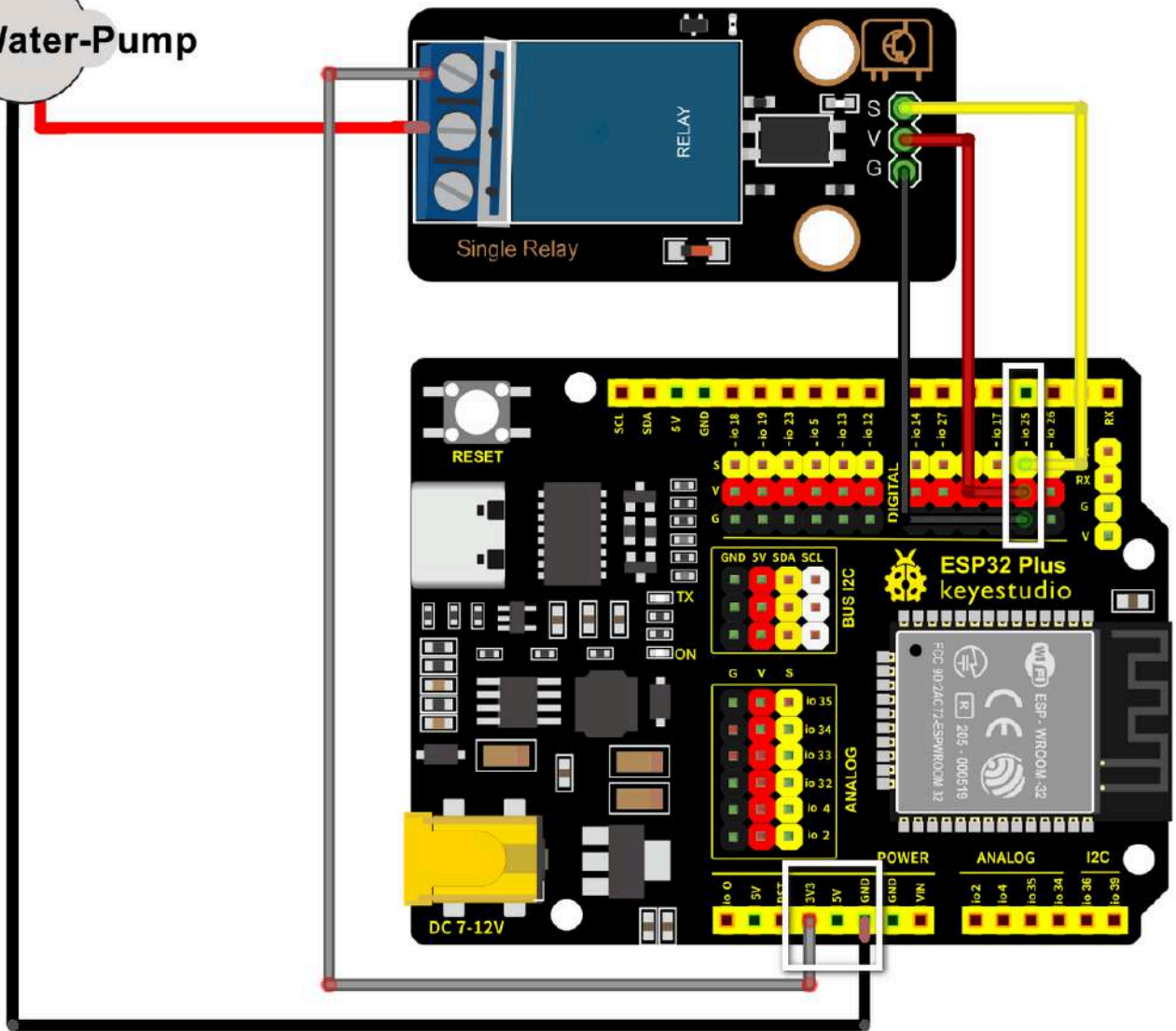
Treceți firul pompei de apă prin orificiul 11 în modul indicat sub:



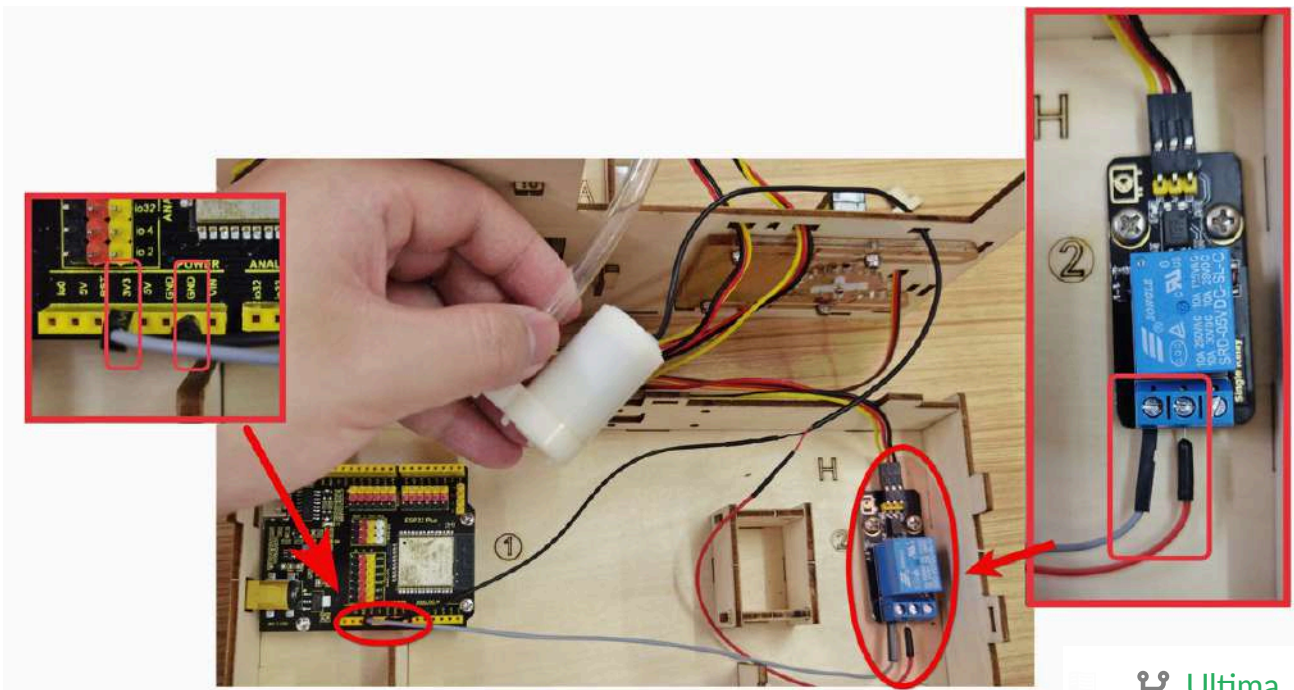
Firul roșu al pompei de apă este conectat la terminalul din mijloc al modulului de releu, iar firul negru este conectat la GND al Placă ESP32.

În plus, trebuie să utilizați un fir Dupont pentru a conecta terminalul din stânga modulului releu la 3,3 V al ESP32.

Water-Pump



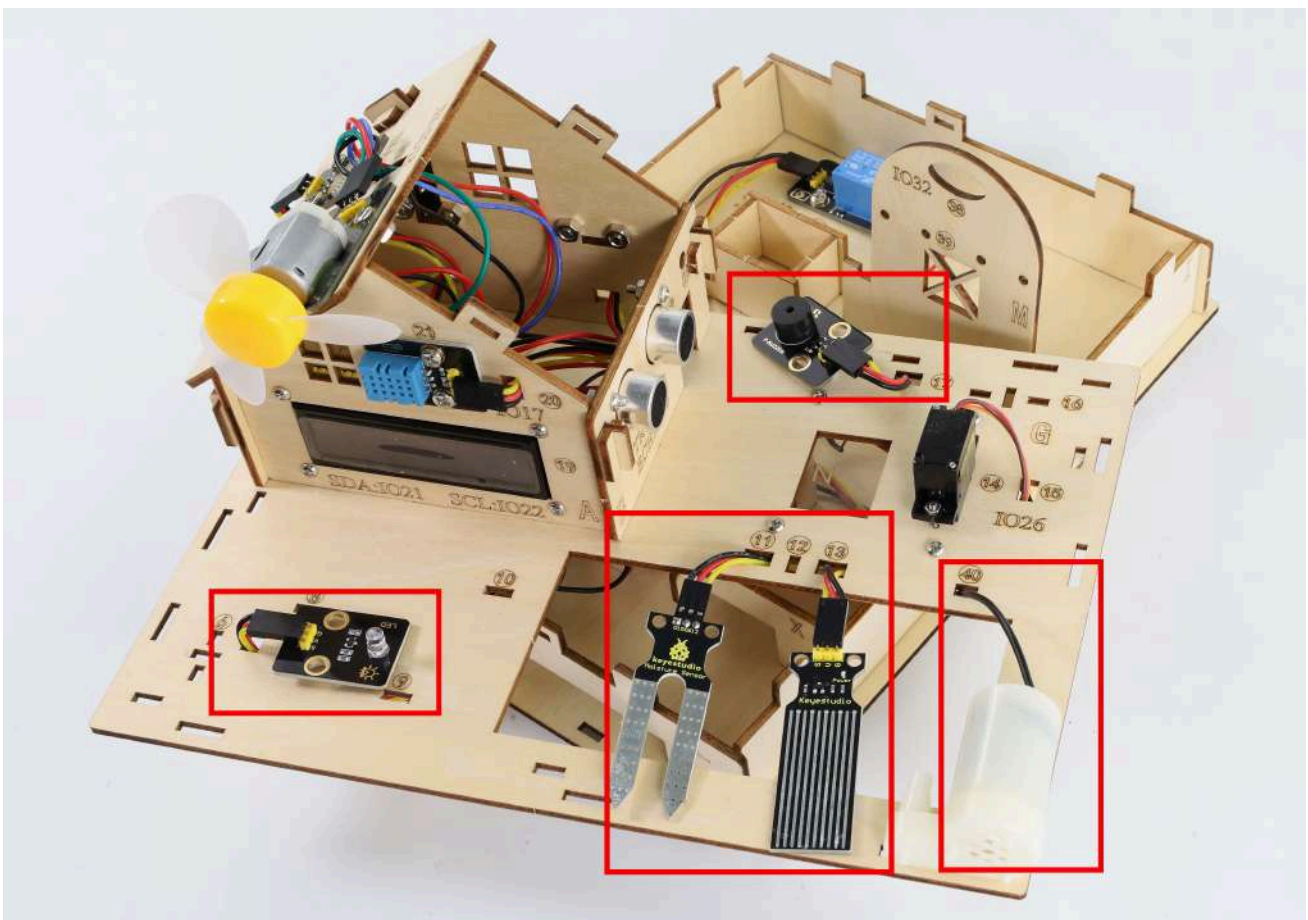
11.16



Introduceți terminalul tată al firului Dupont în terminalul mamă al modulul de releu și strângeți-l cu o șurubelniță.

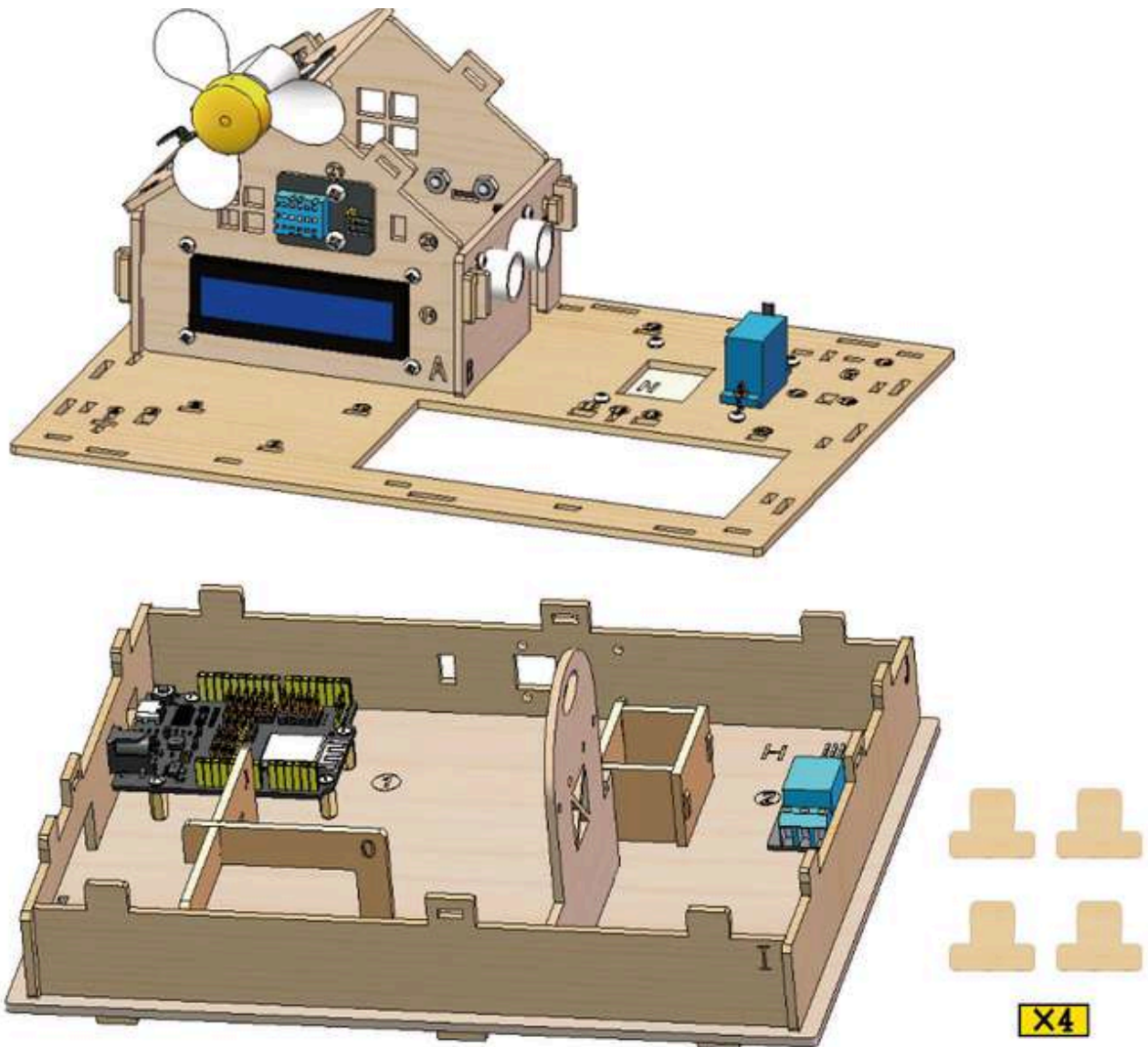


După pașii de mai sus, rețineți că toate cablajele au fost finalizate. Și firele LED-ului, senzorul de nivel al apei, senzorul de umiditate a solului, soneria și pompa de apă releu au trecut prin găurile de 7, 11, 13, 17 și 40 sculptate pe placa de tei, pregătindu-se pentru următoarele instalare.

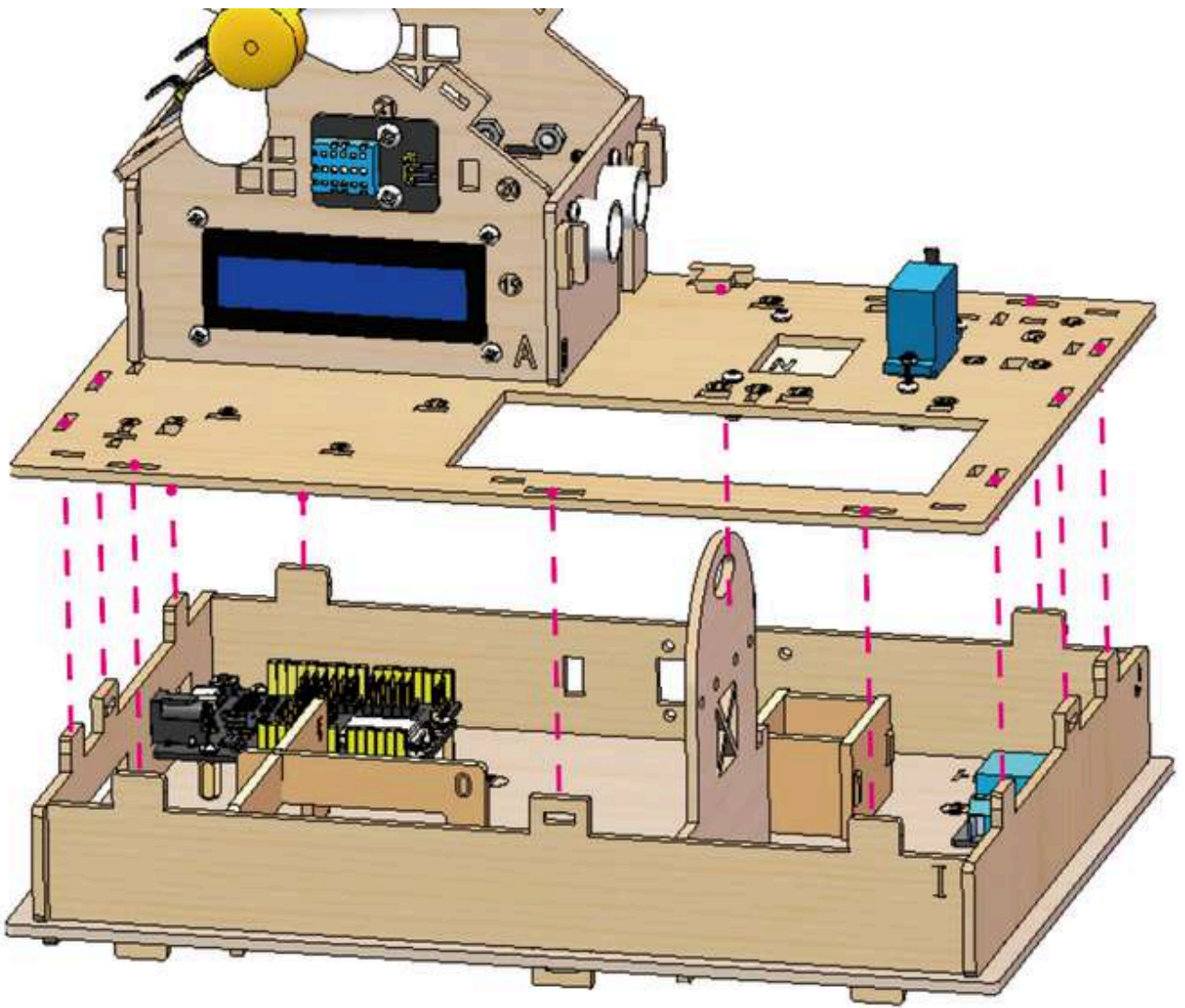


Pasul 12 Instalați casa și fundația

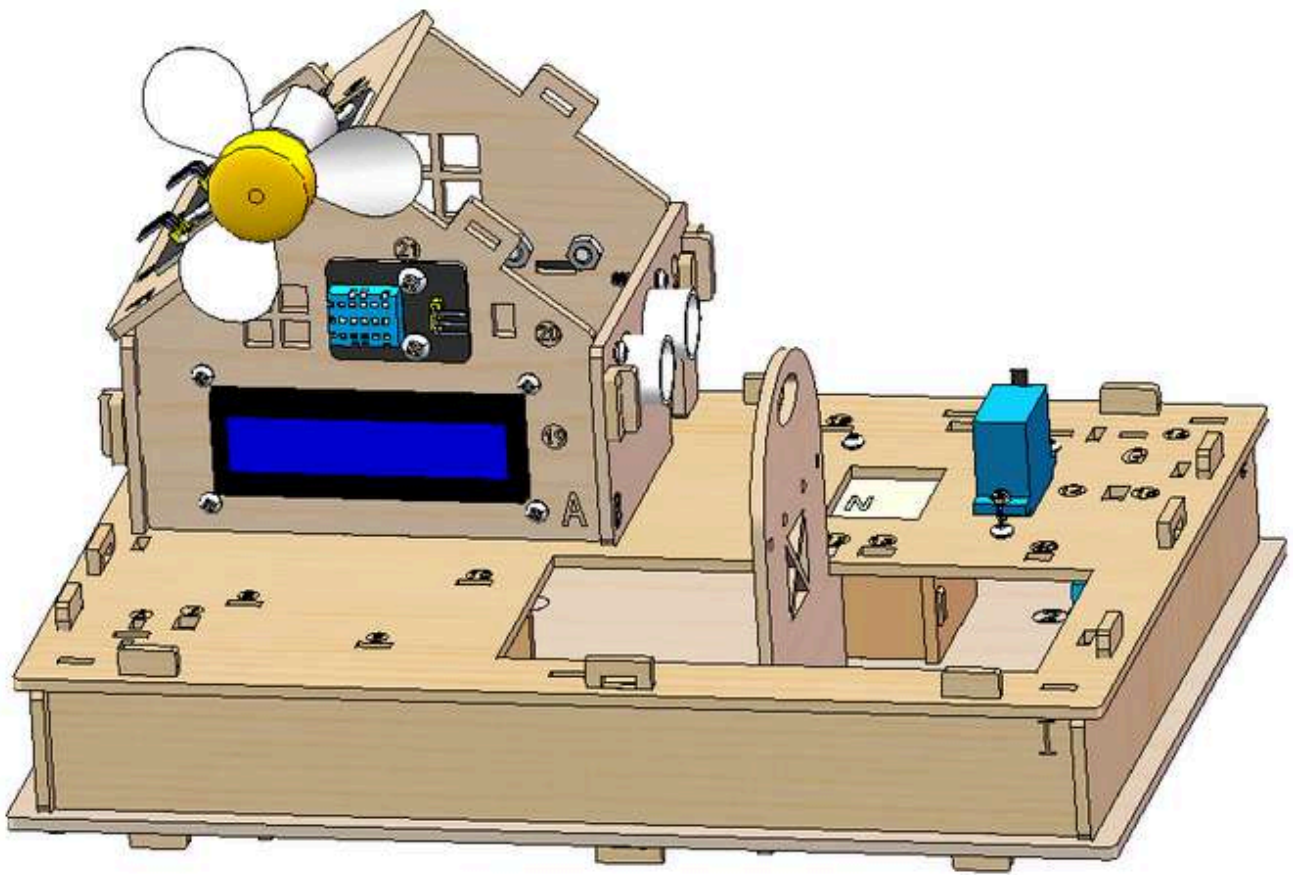
12.1 Componente necesare



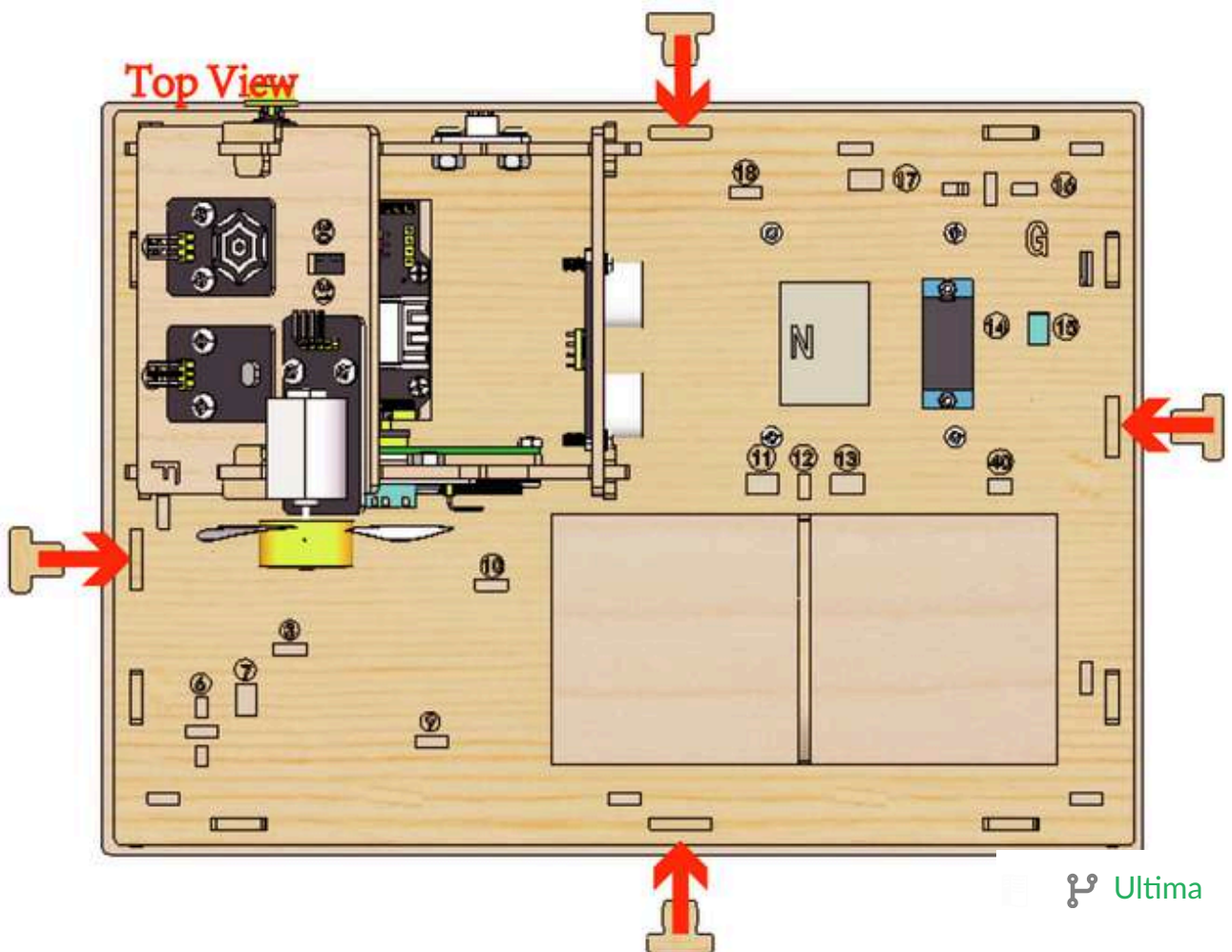
12.2

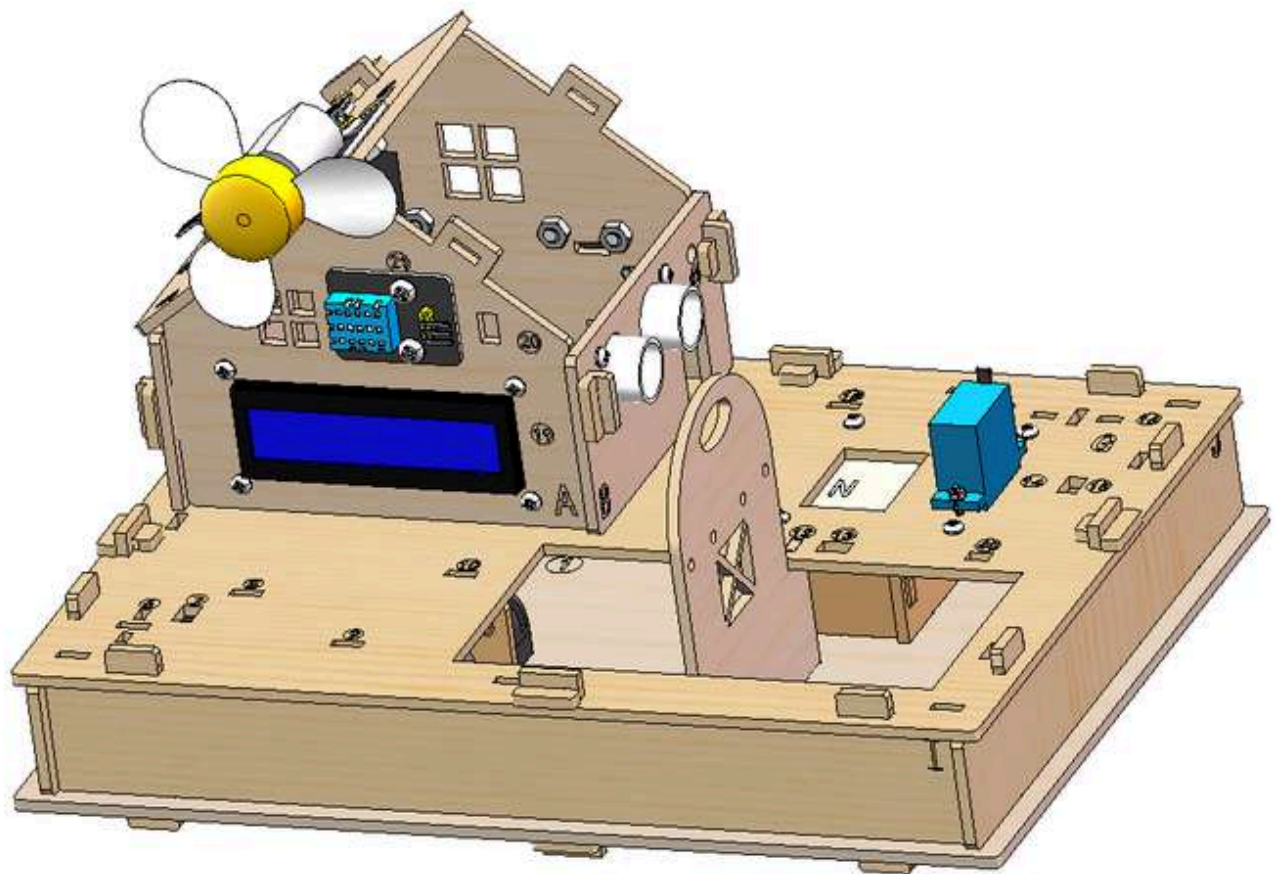


12.3



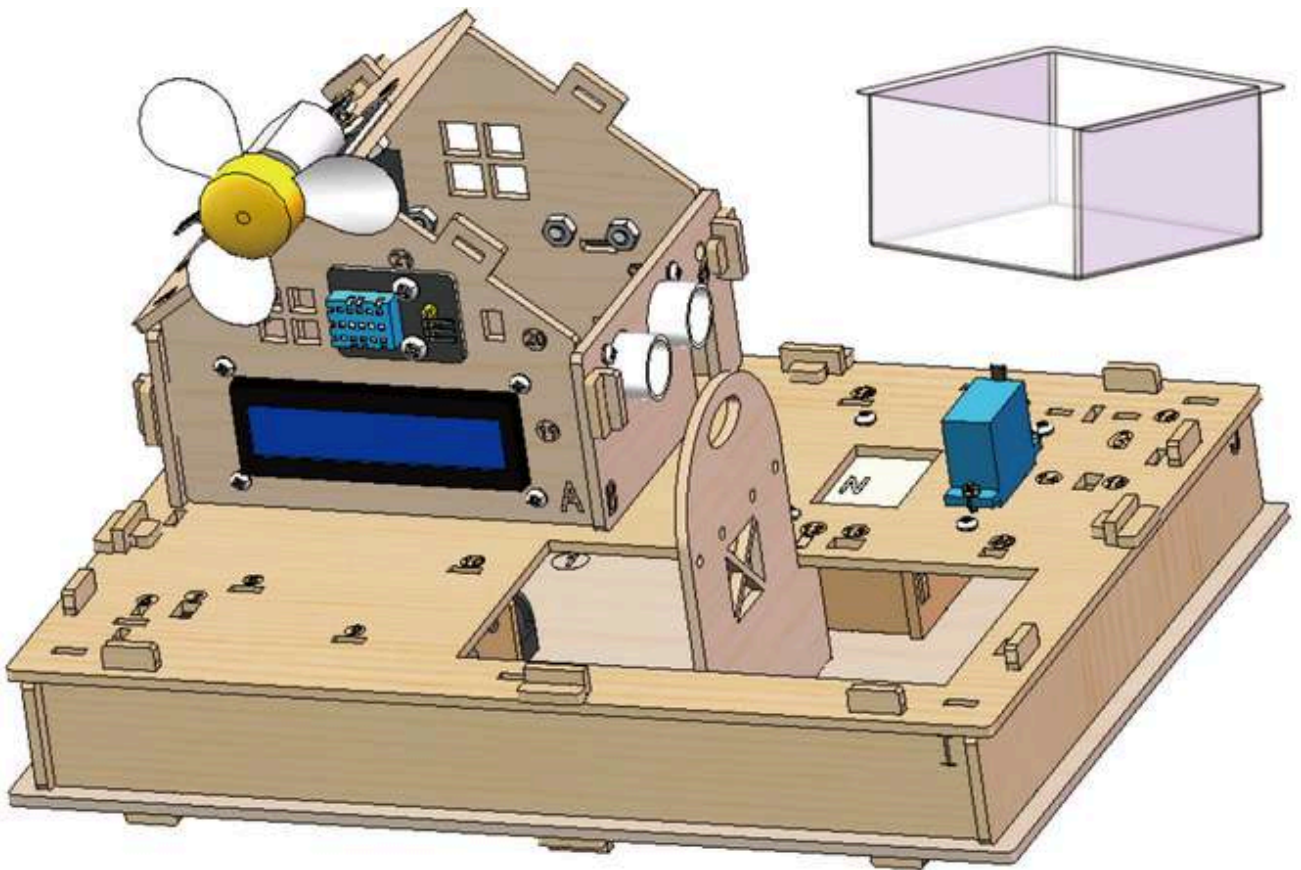
12.4



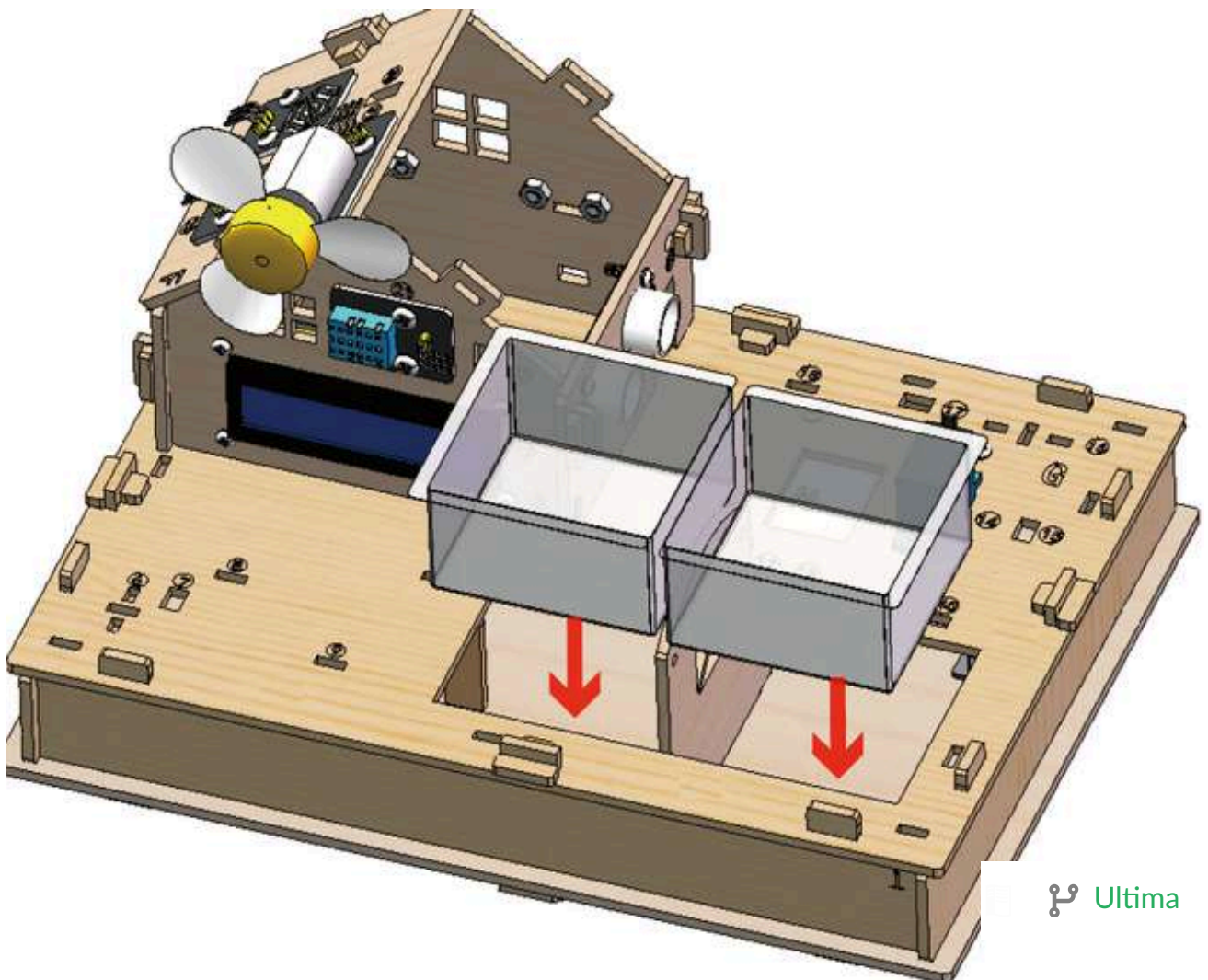


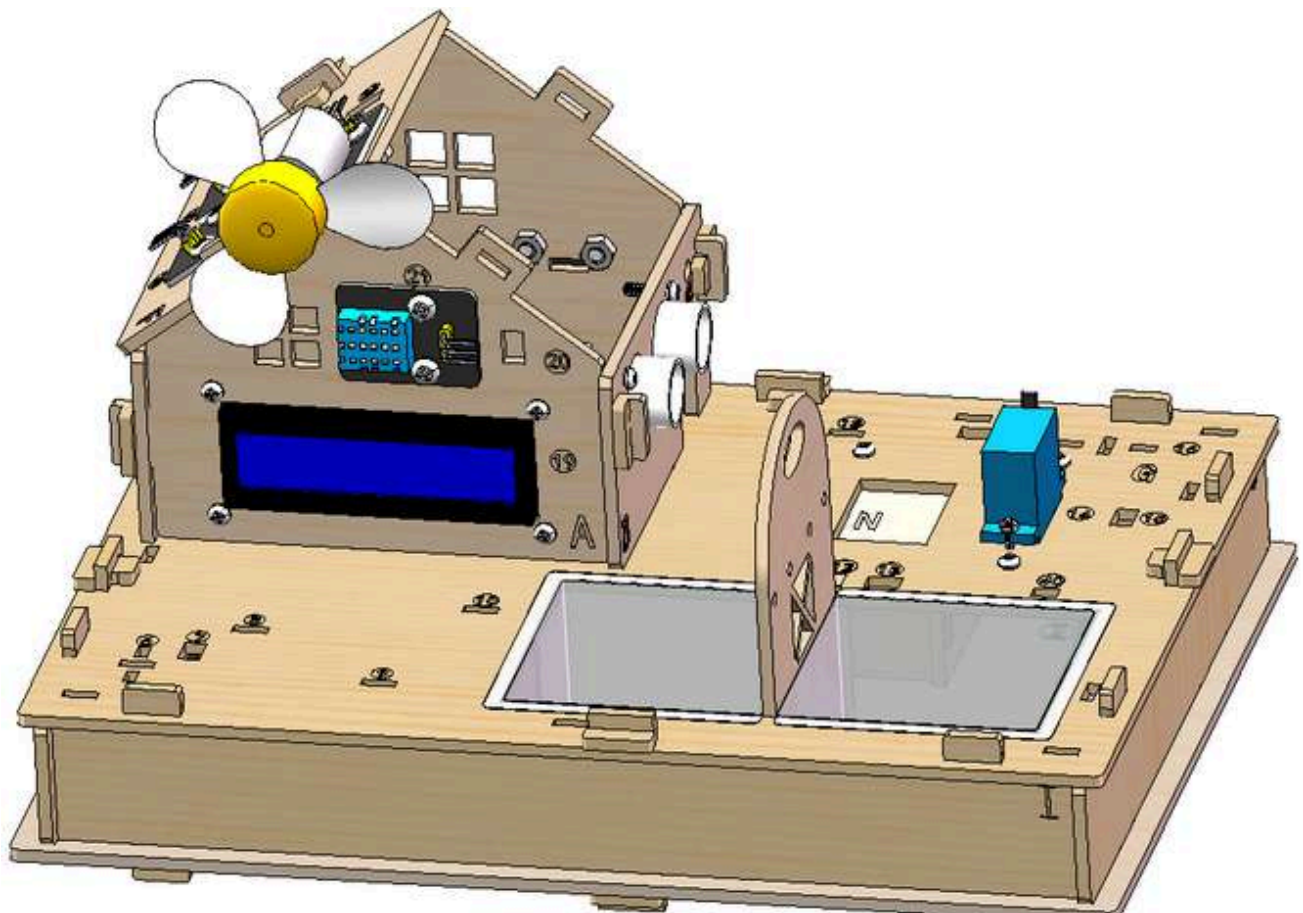
Pasul 13 Instalați chiuvetele din plastic

13.1 Componente necesare



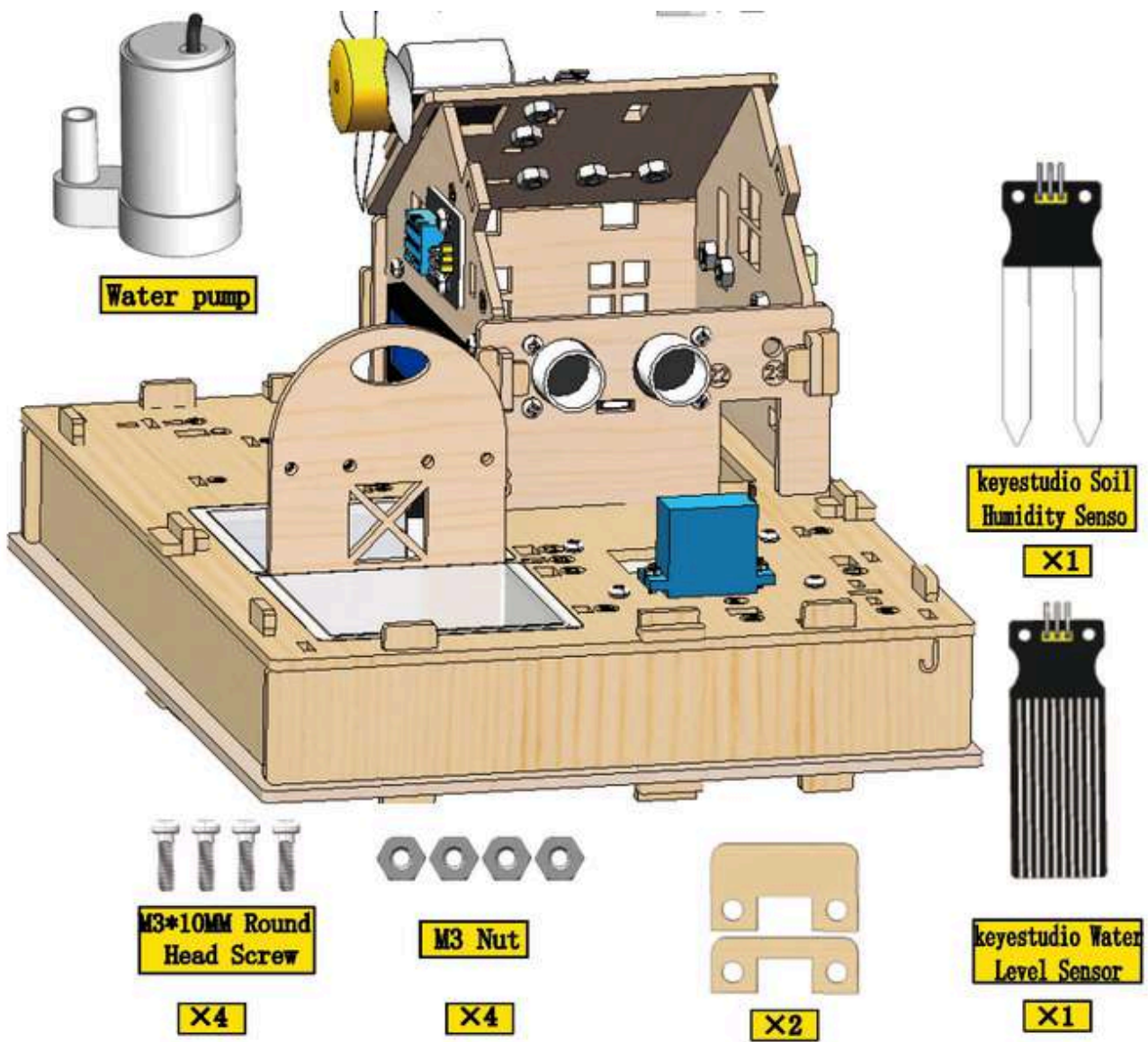
13.2



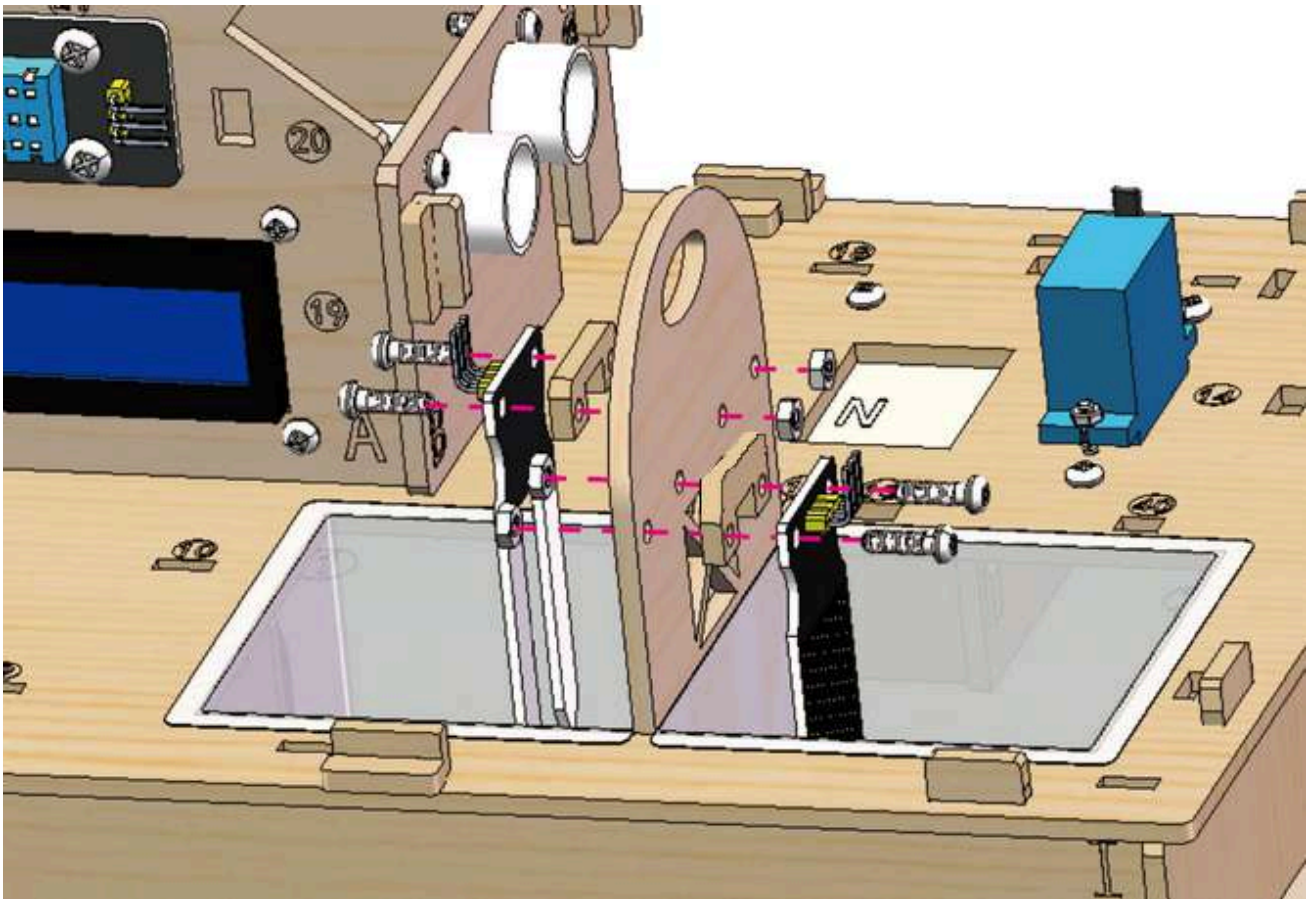


Pasul 14 Instalați modulul de sol și modulul de nivel al apei

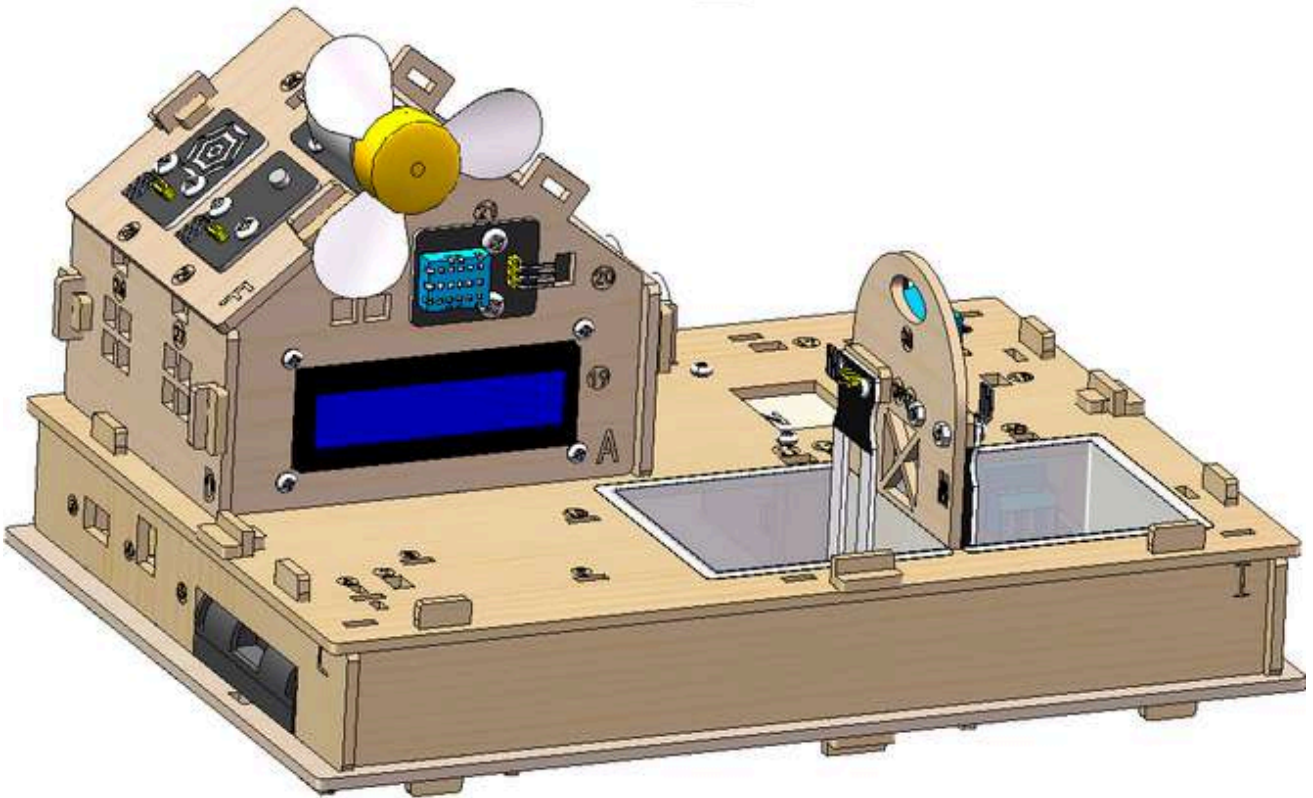
14.1 Componente necesare



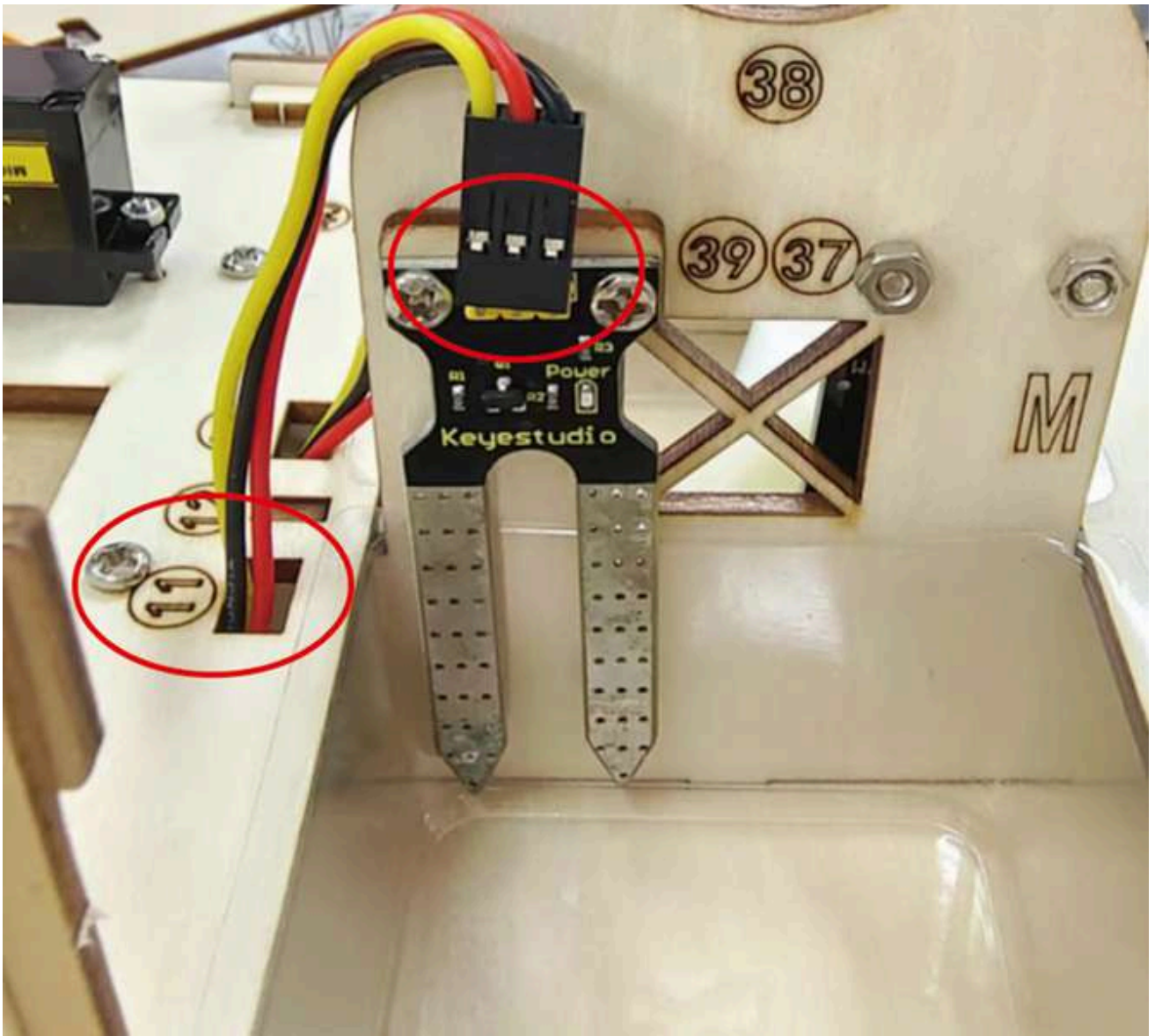
14.2



14.3



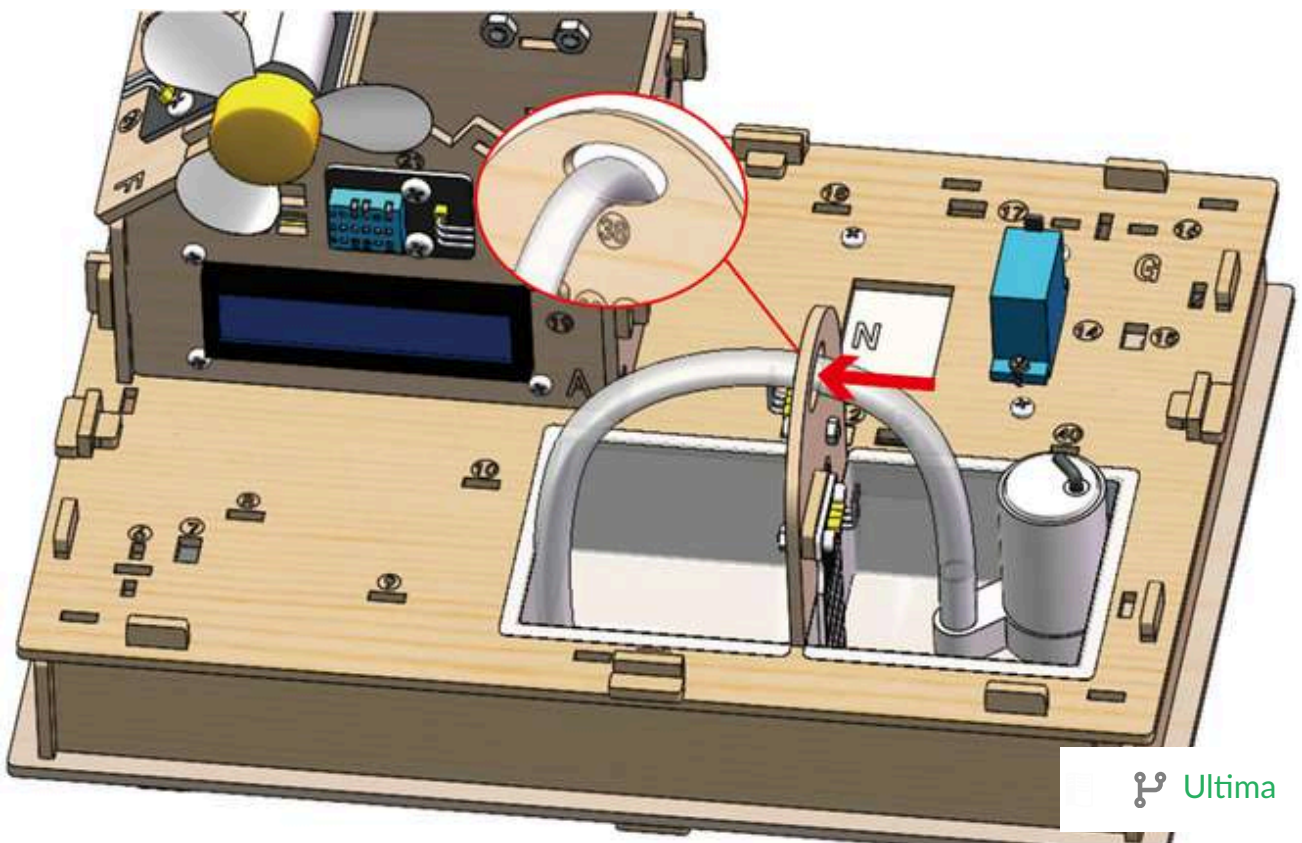
14.4

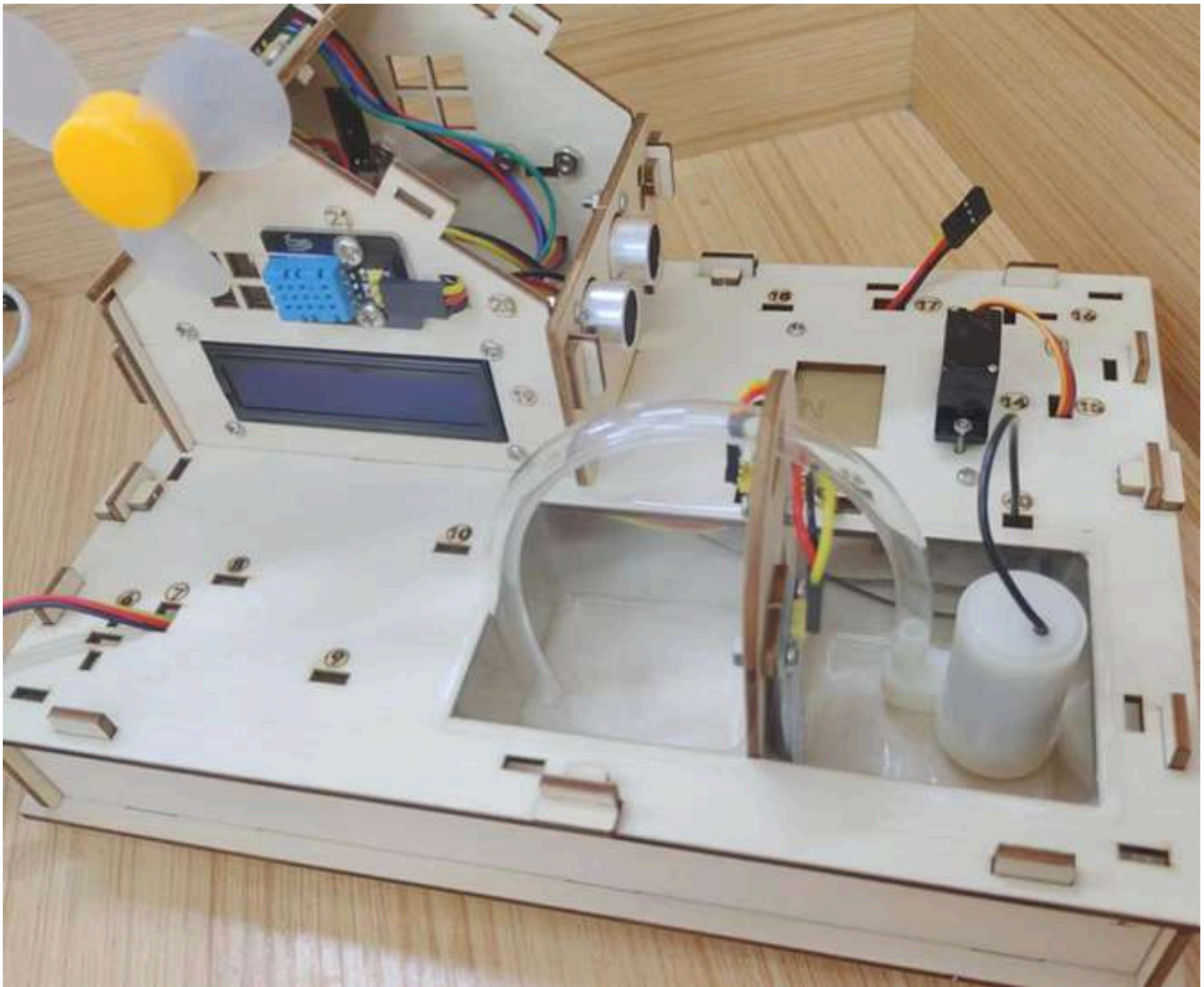


14.5



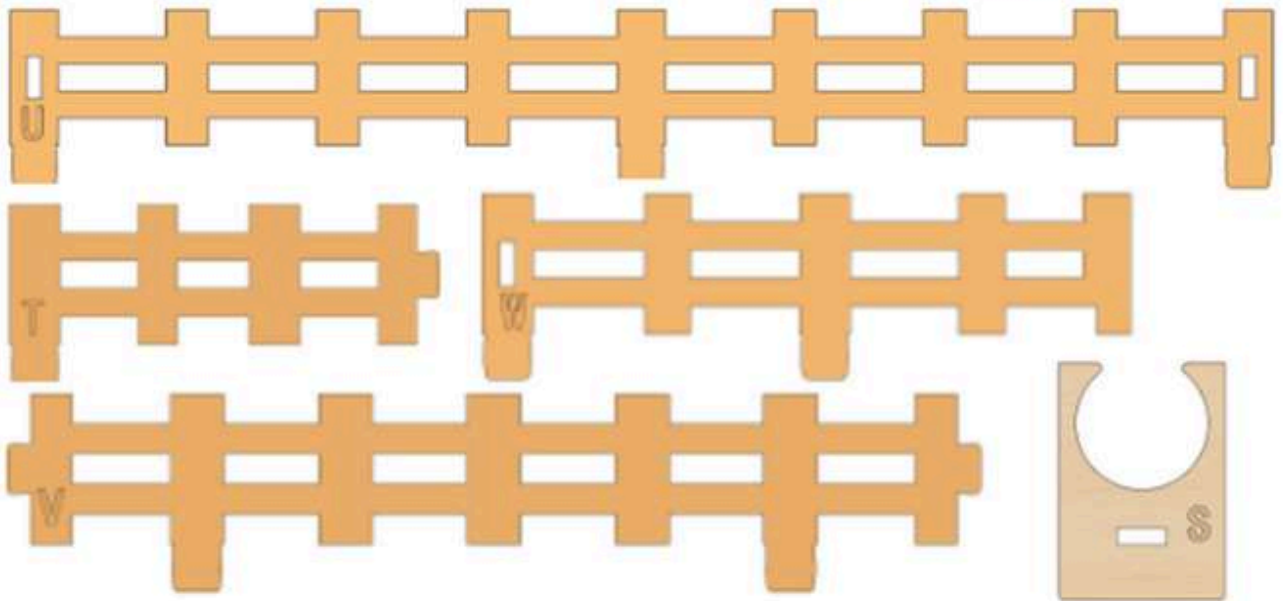
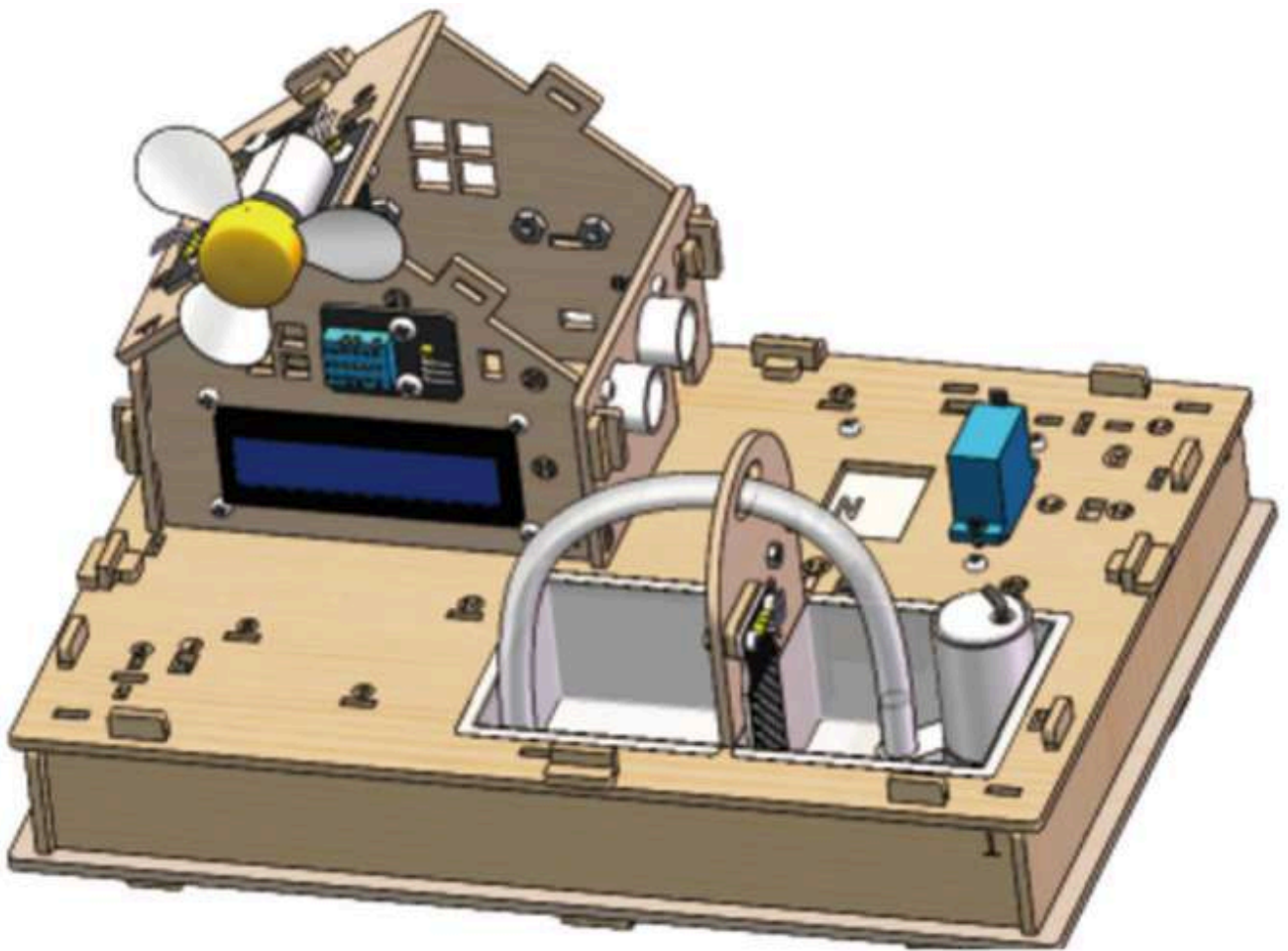
14.6



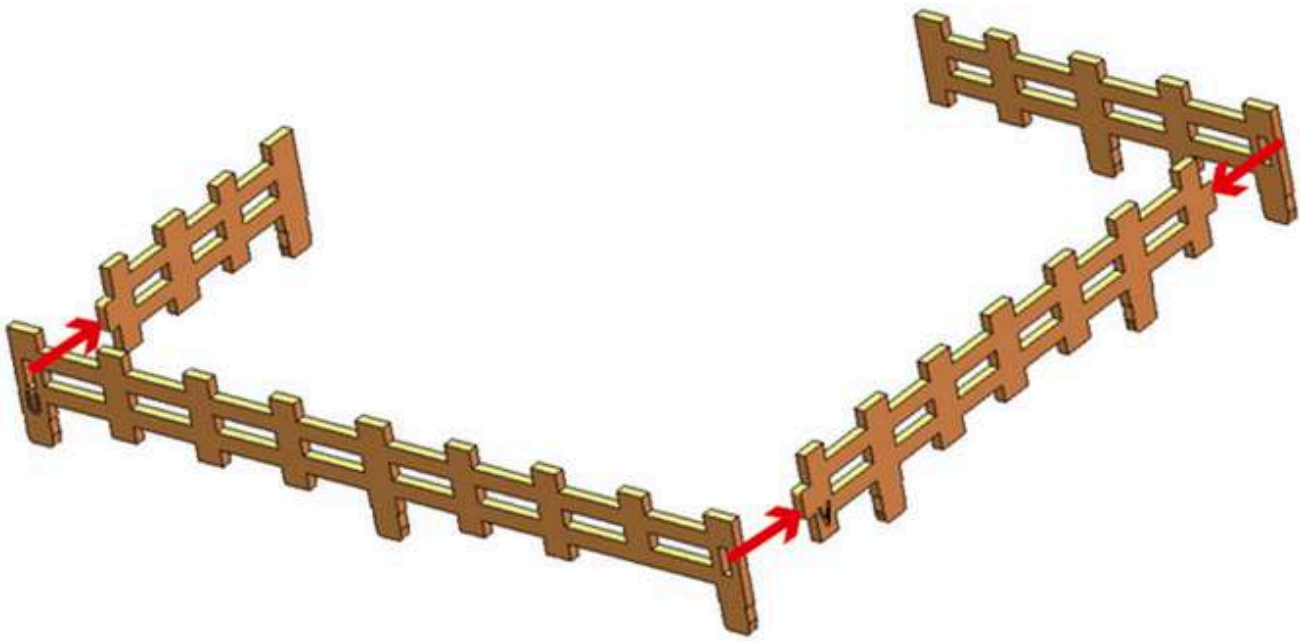


Pasul 15 Instalați gardul

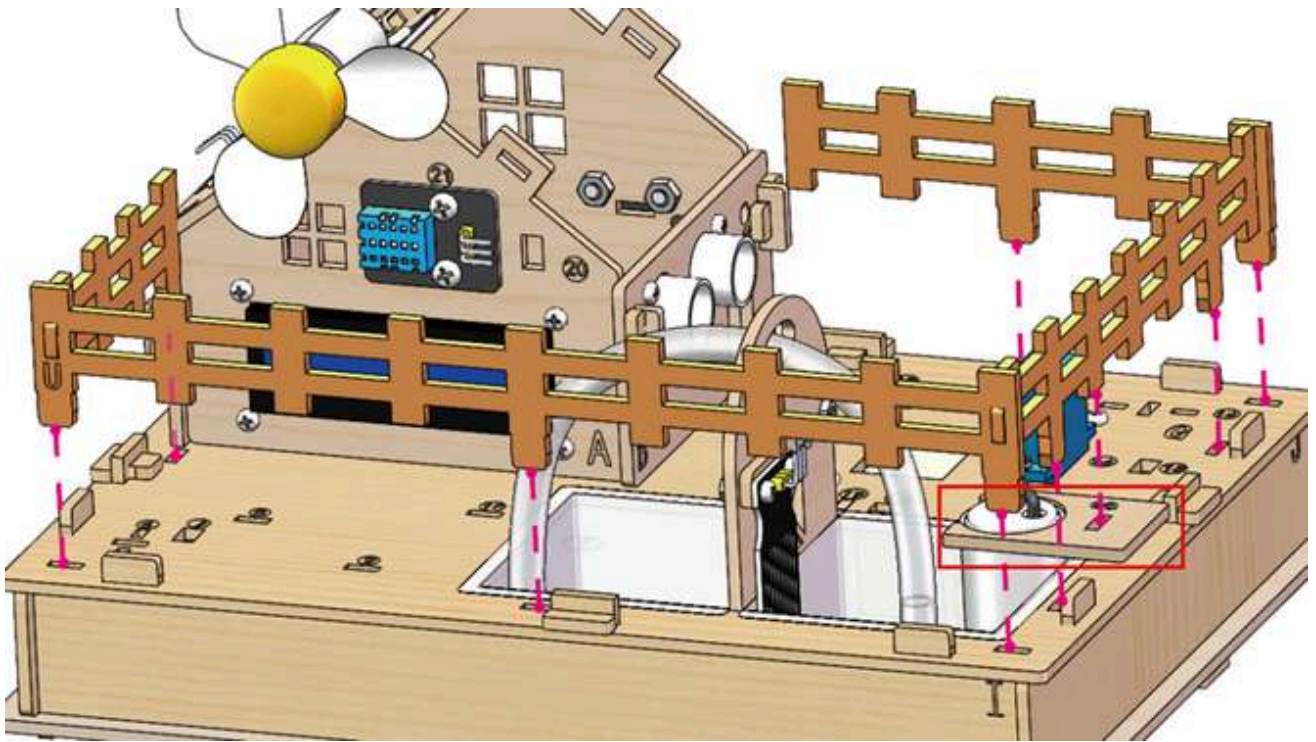
15.1 Componente necesare



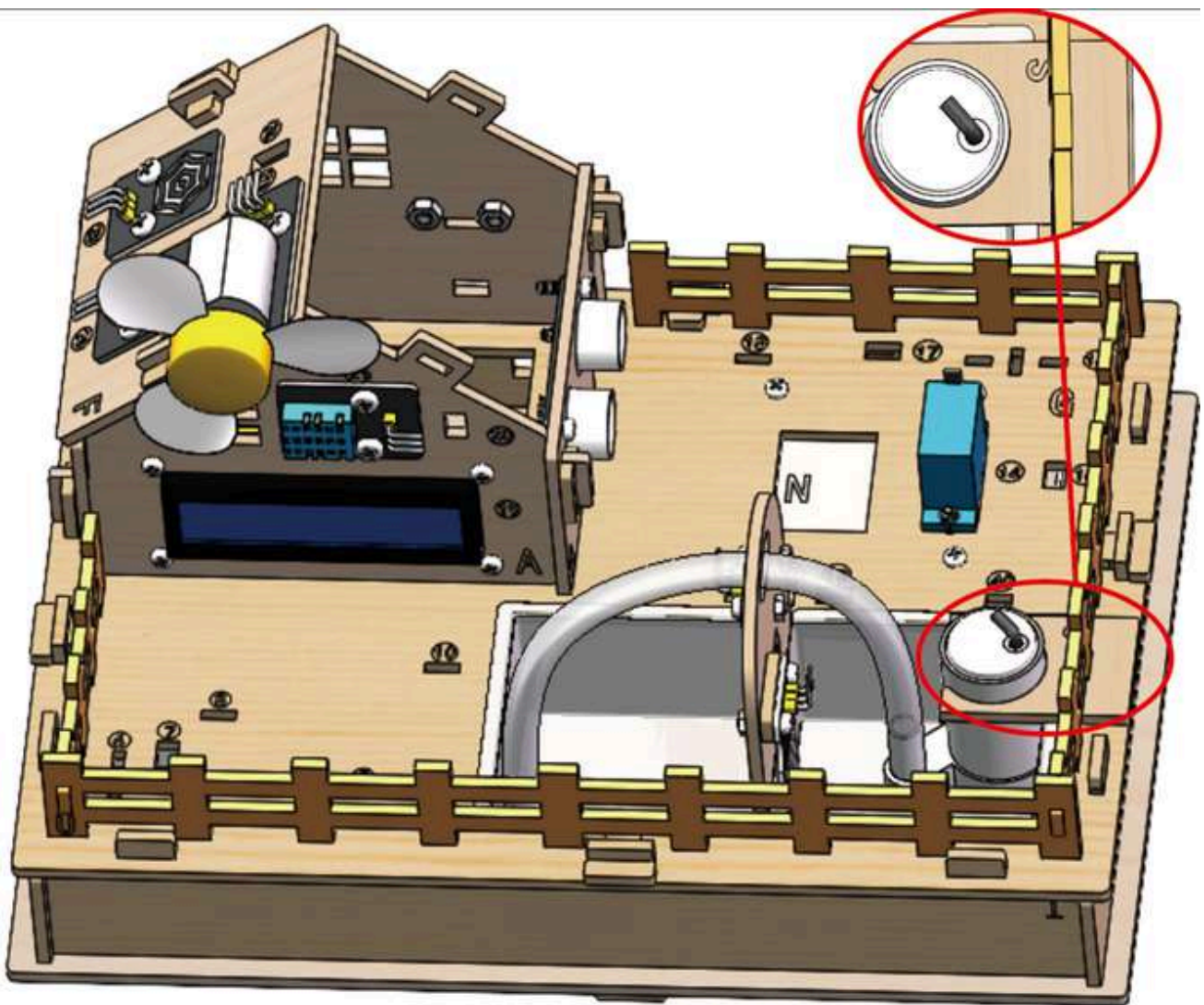
15.2



15.3

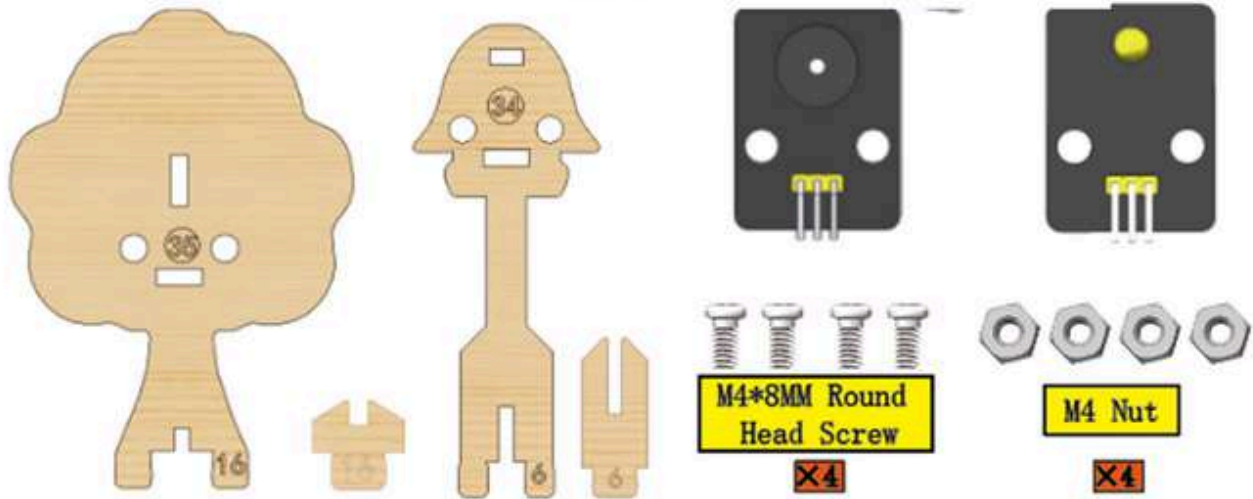
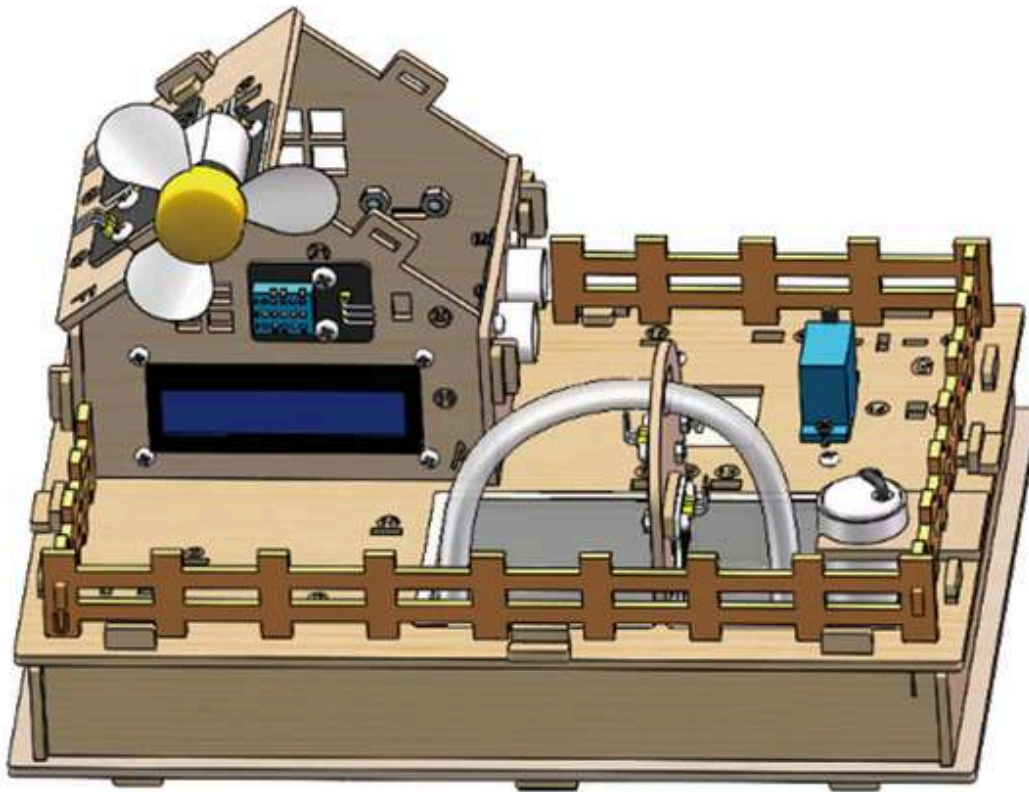


15.4

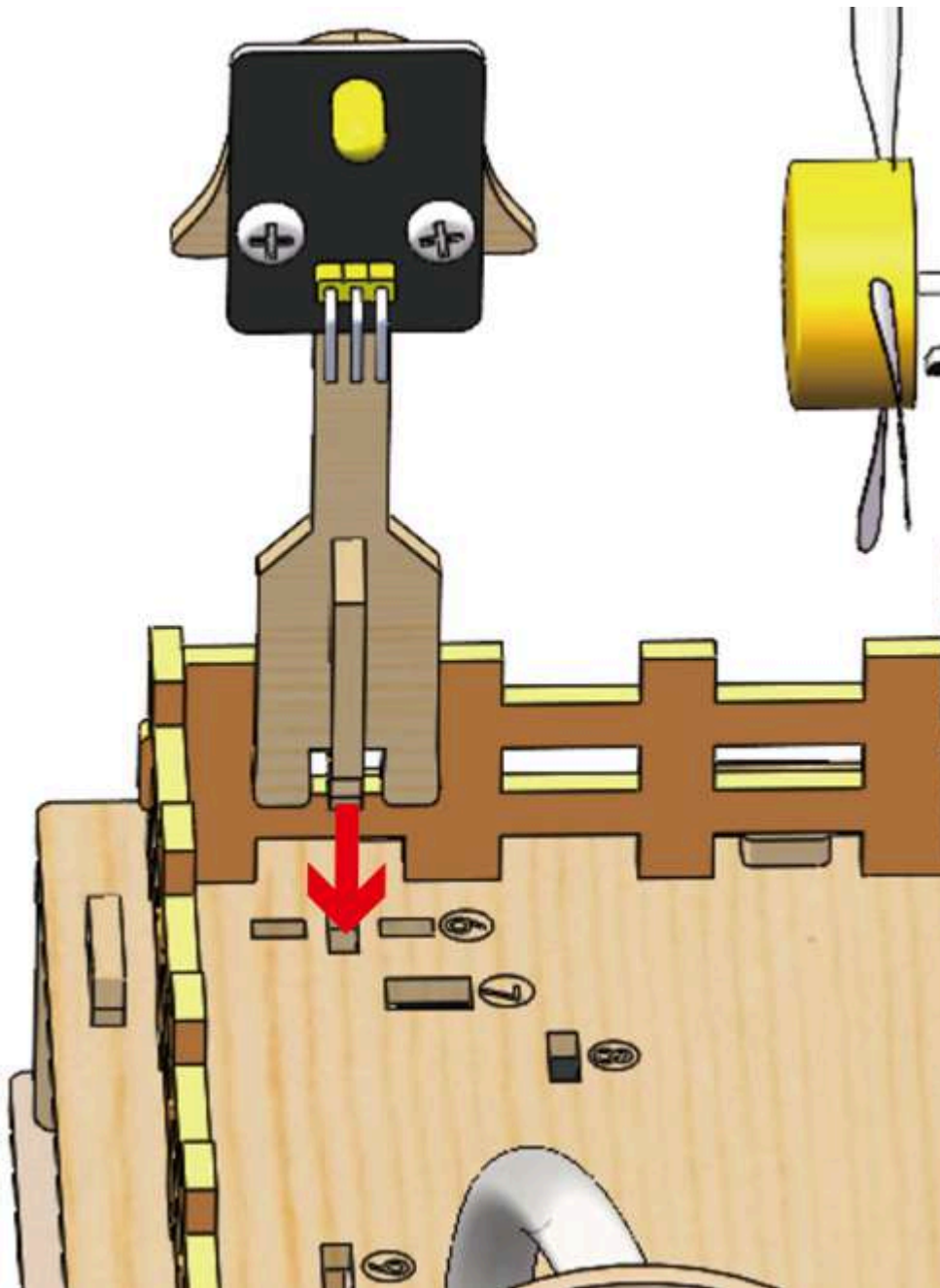


Pasul 16 Instalați soneria și modulul LED

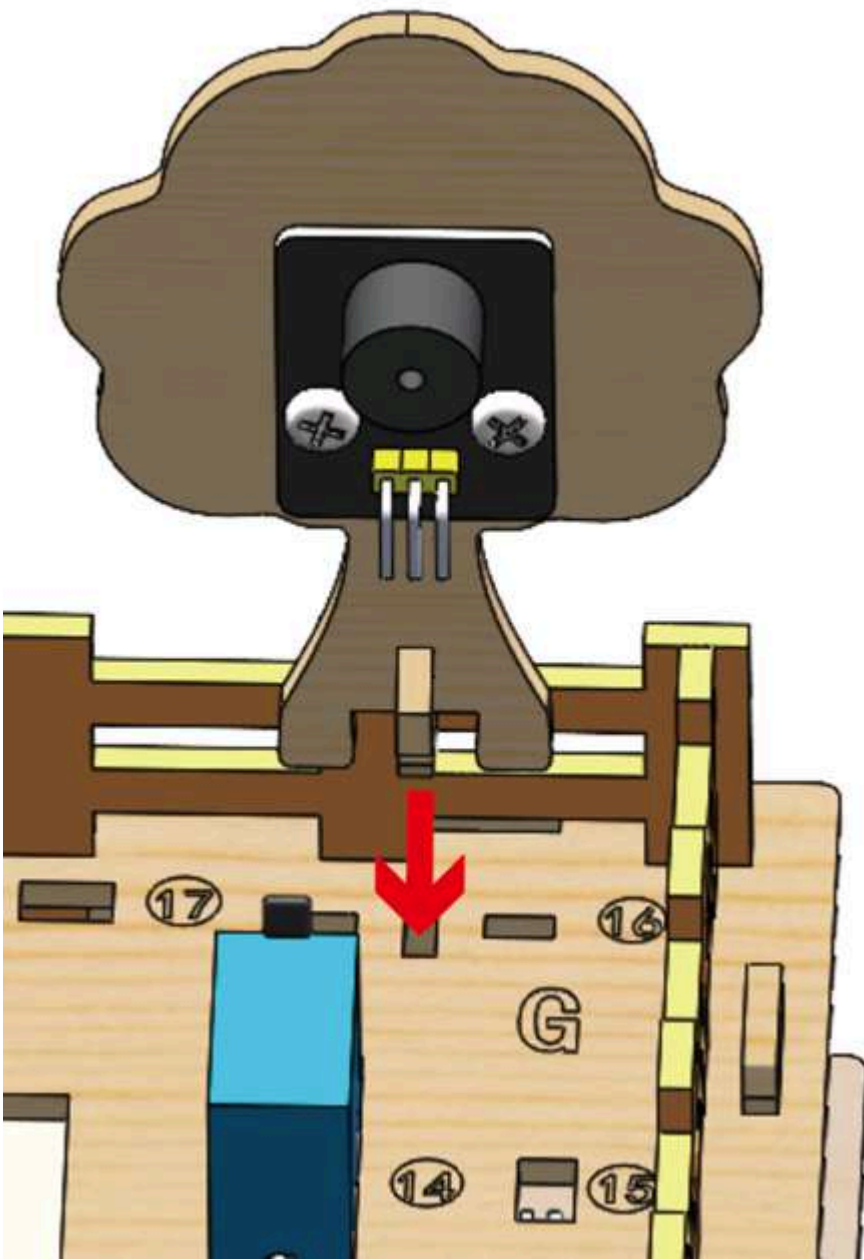
16.1 Componente necesare



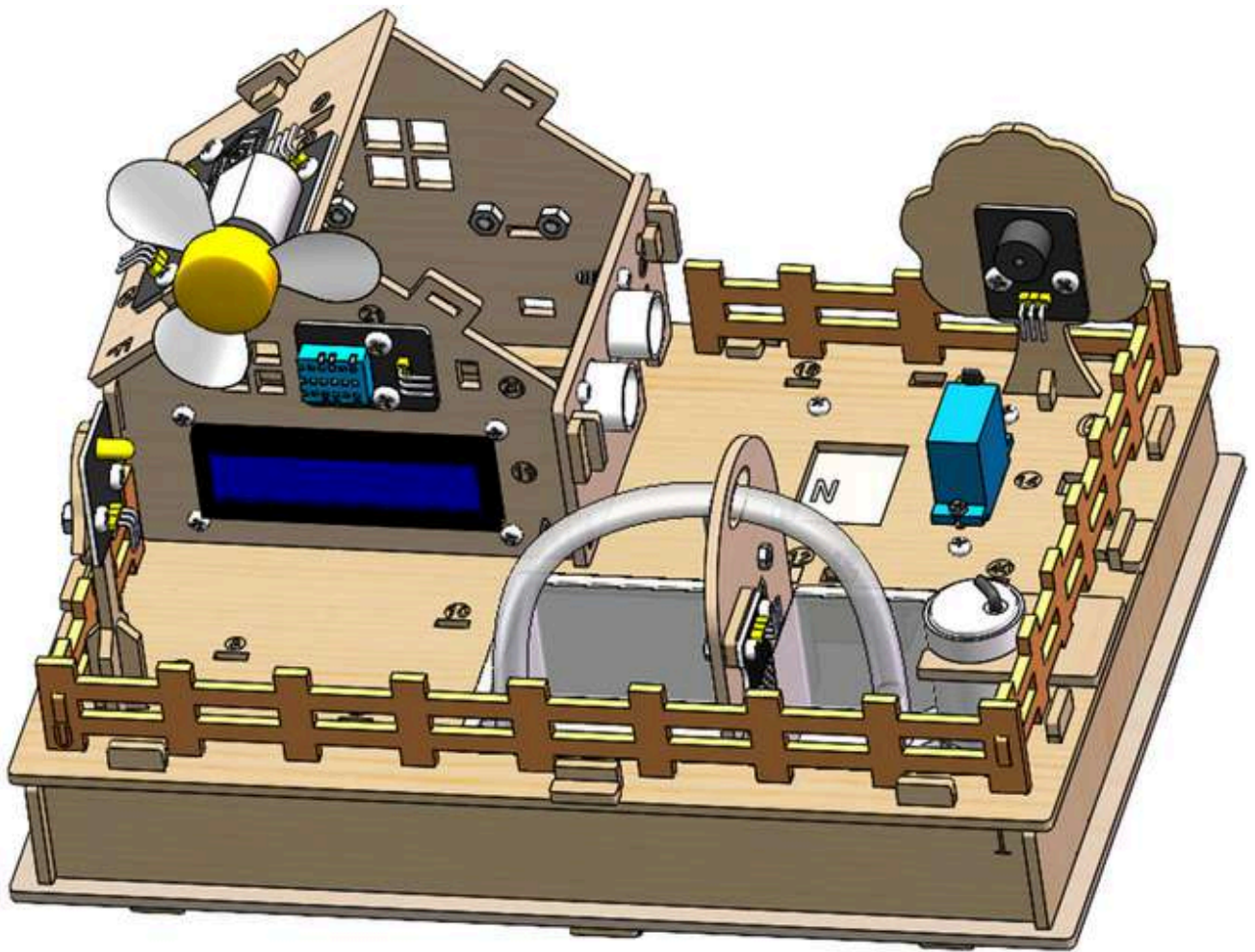
16.2



16.3

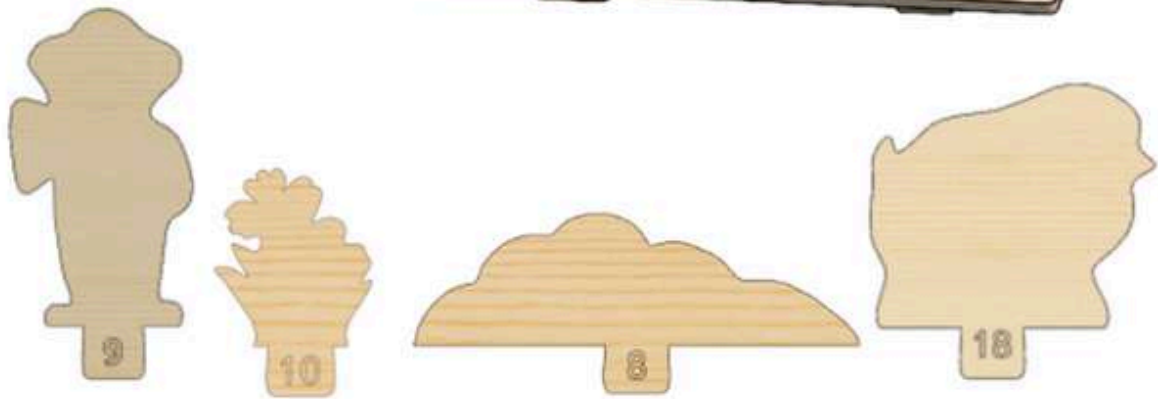
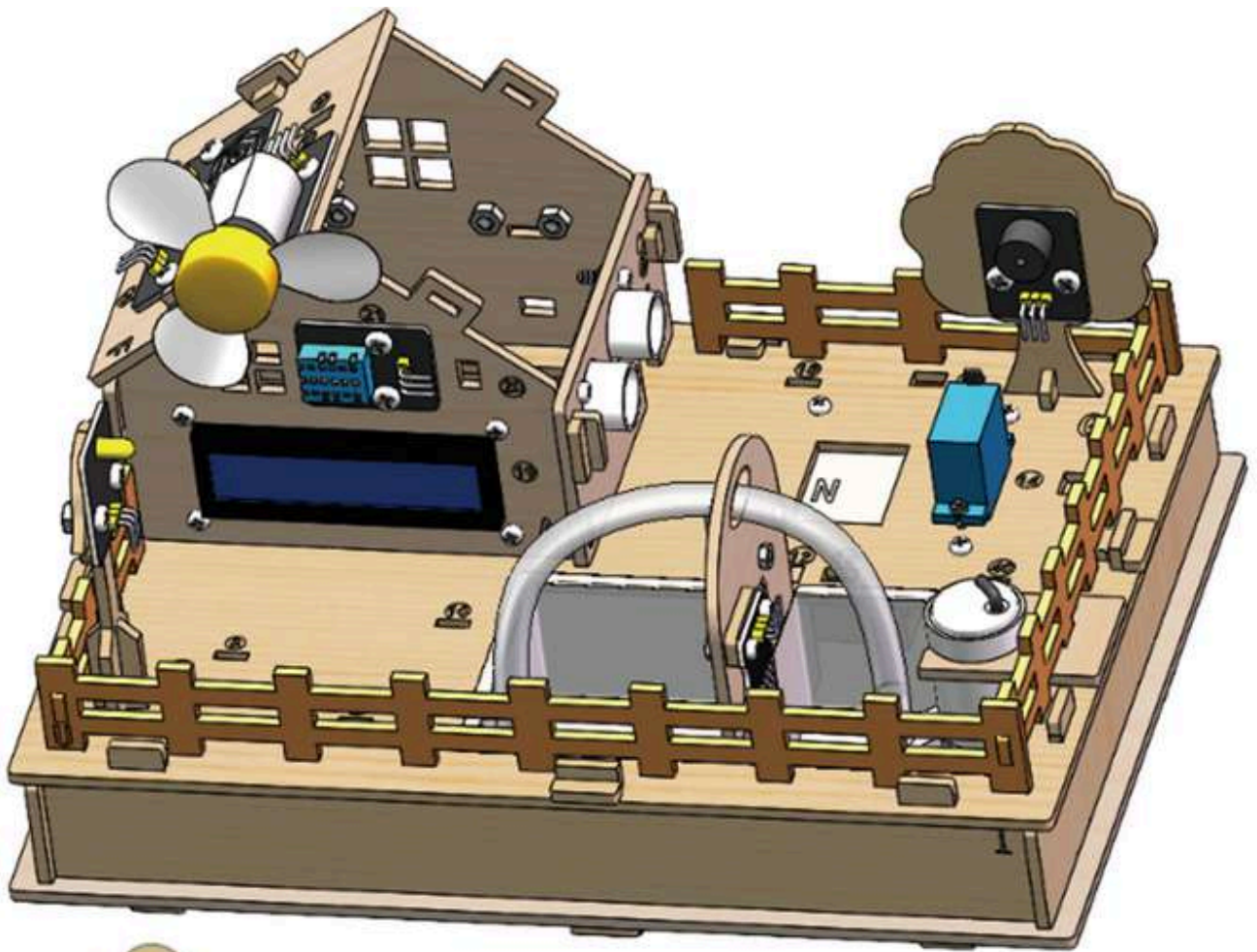


16.4

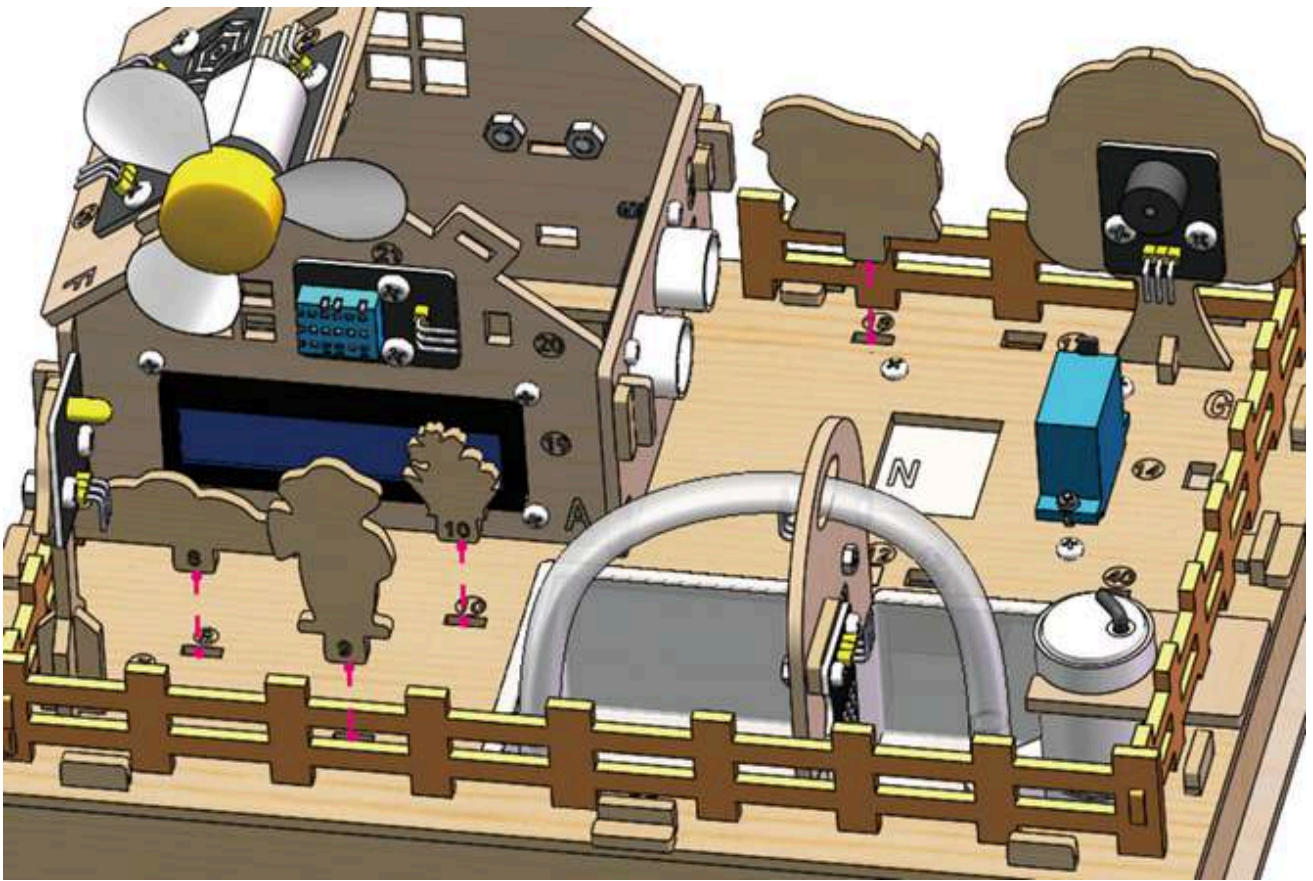


Pasul 17 Decorati casa

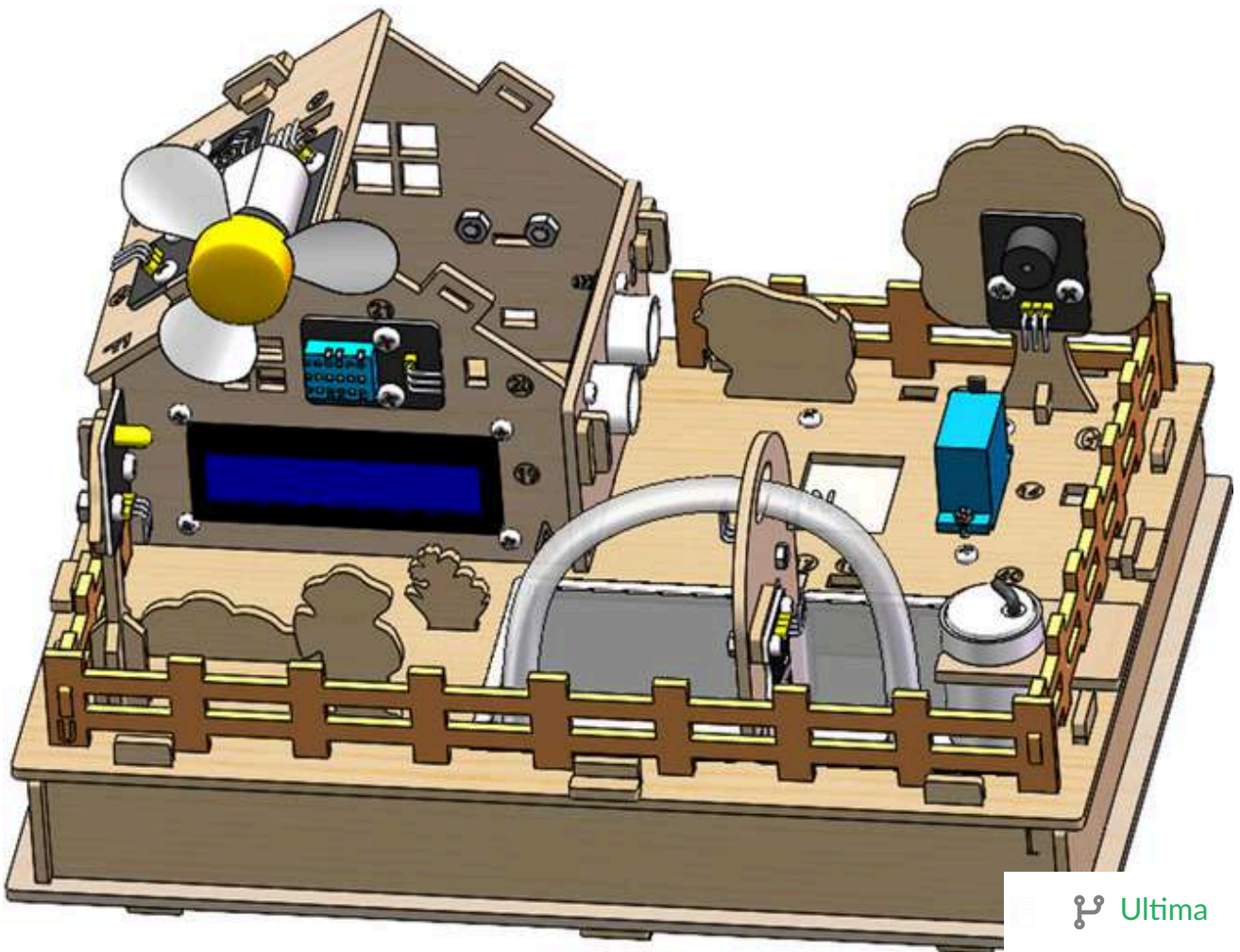
17.1 Componente necesare



17.2

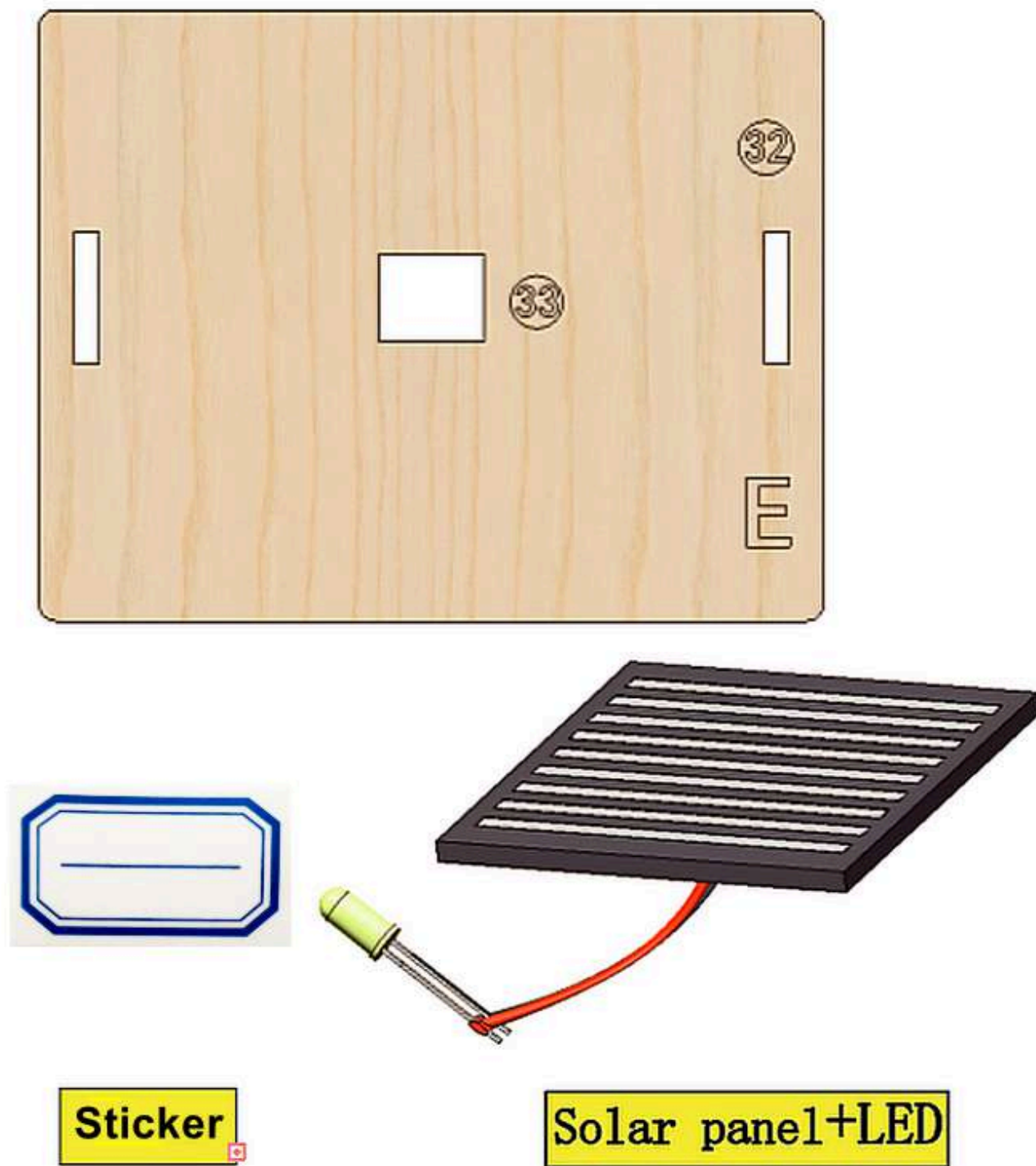


17.3

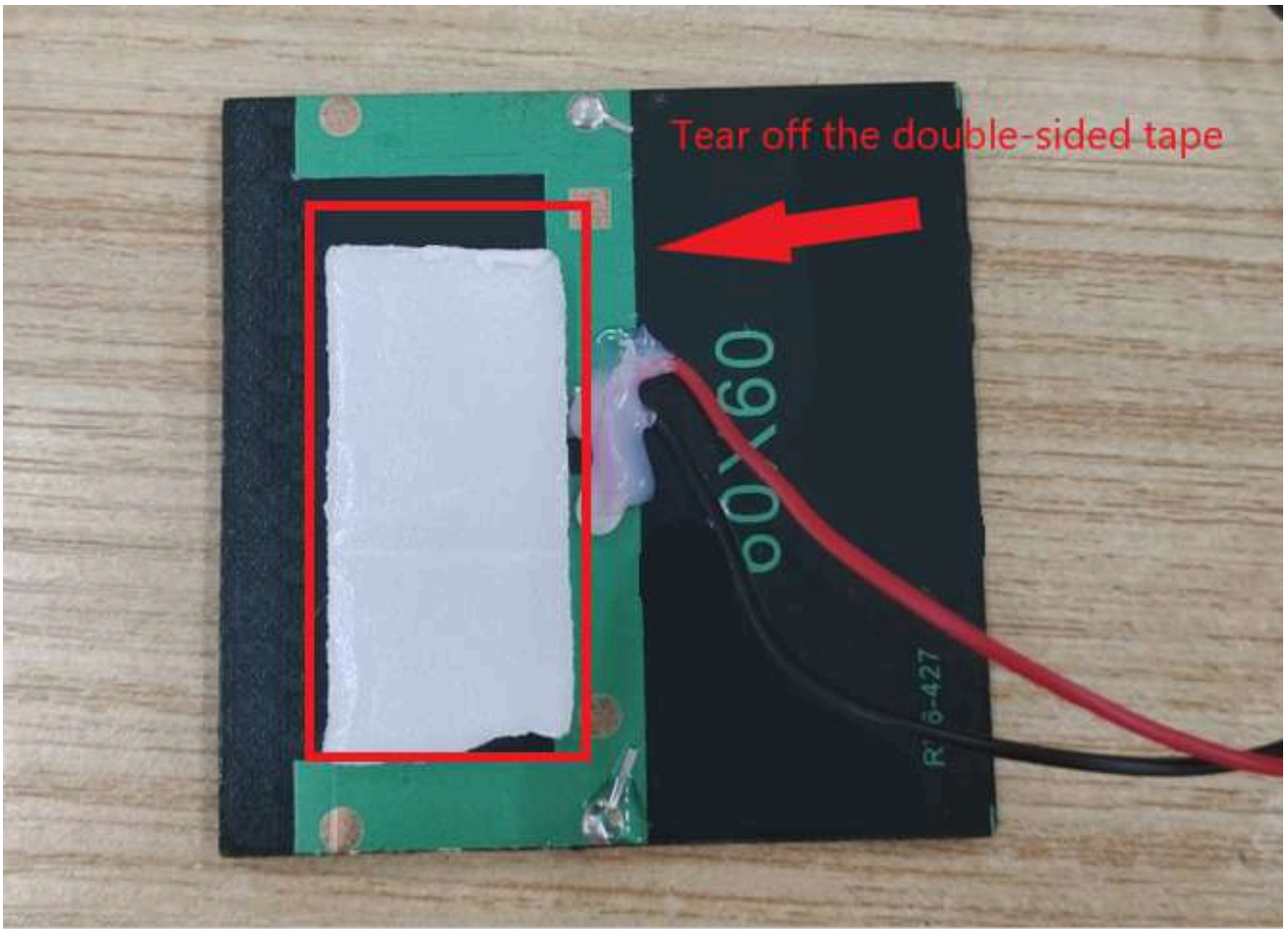


Pasul 18 Instalați panoul solar

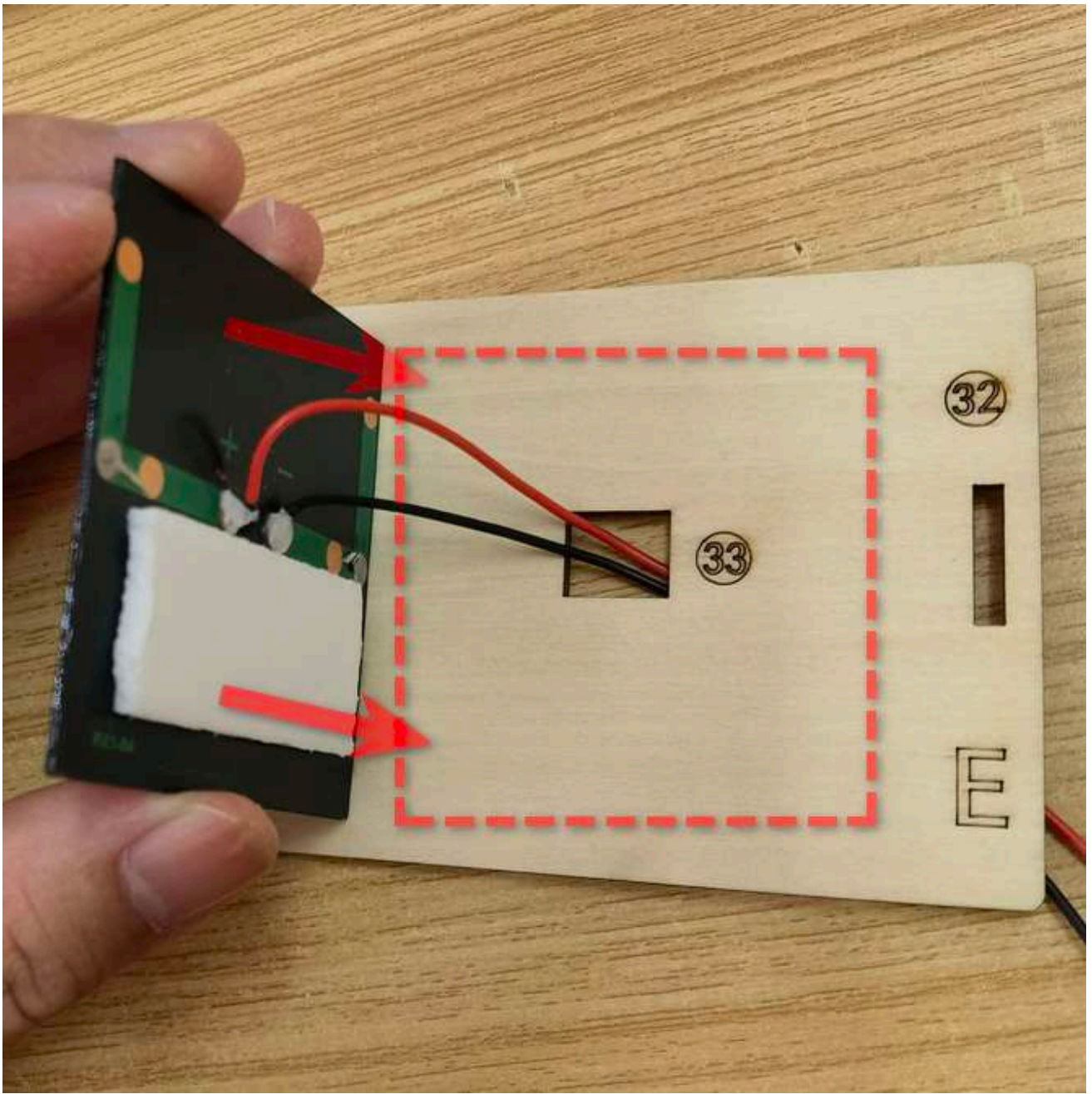
18.1



18.2



18.3

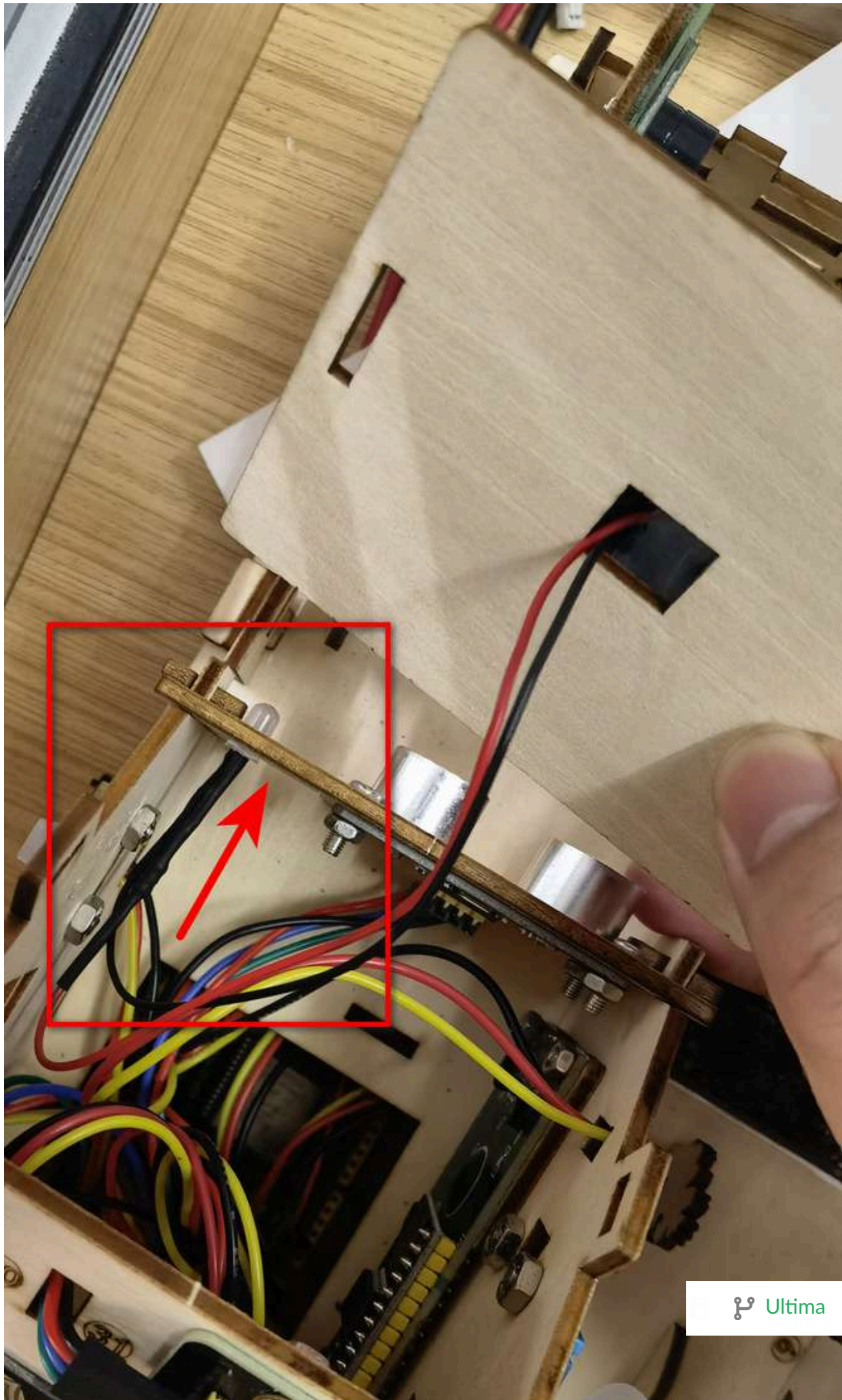


18.4



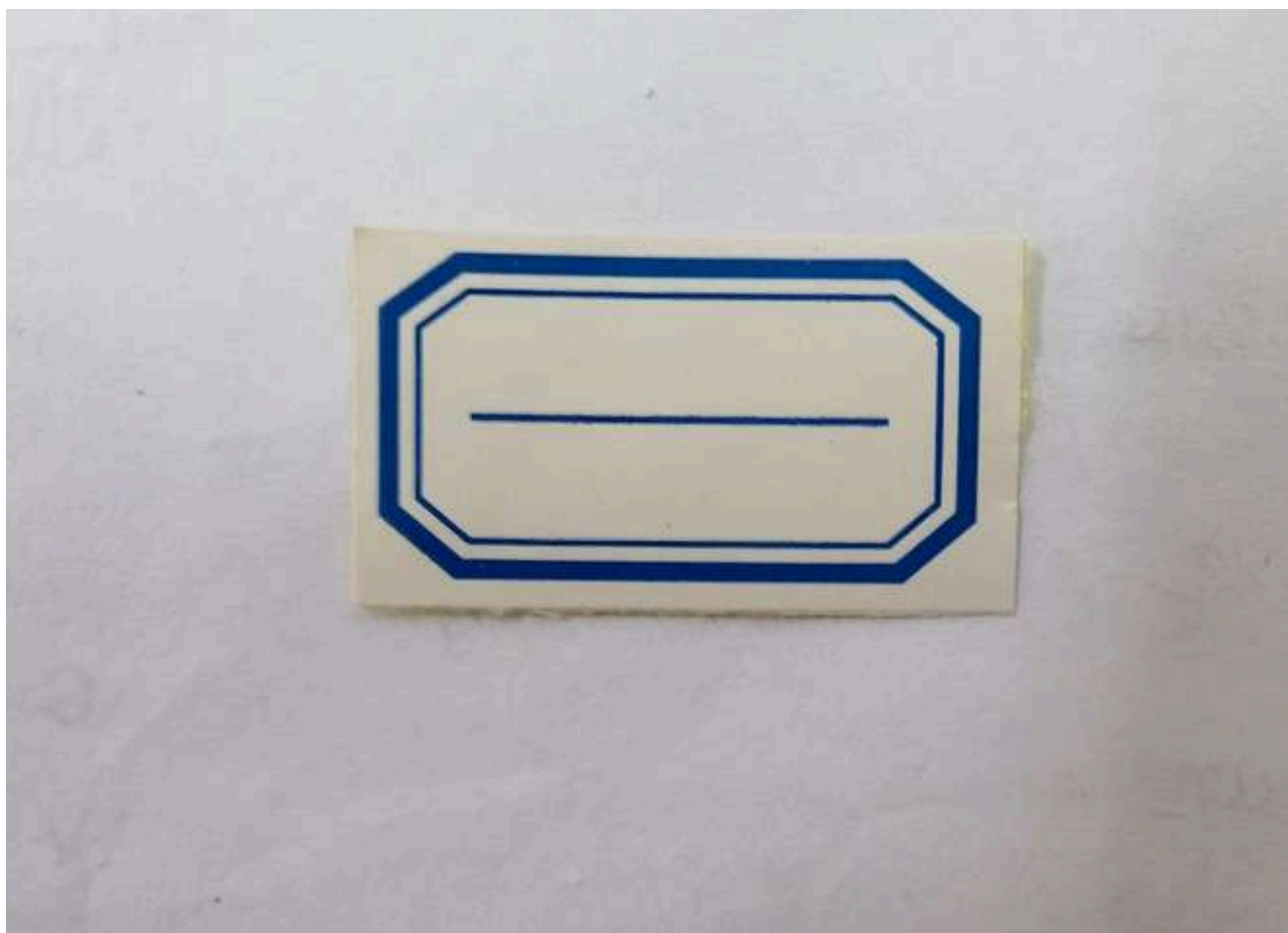
18.5

Instalați lumina LED a panoului solar în această gaură.

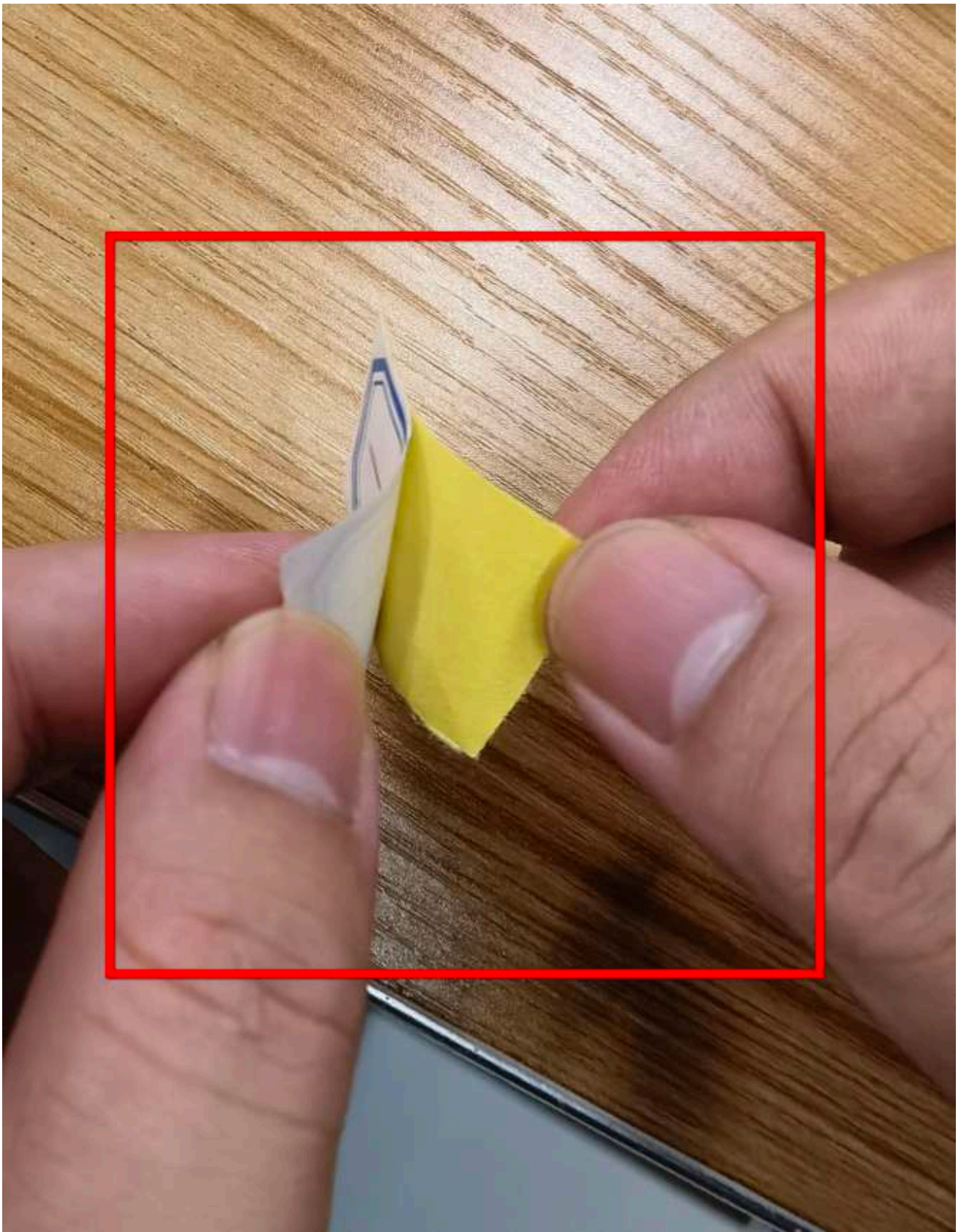


18.6

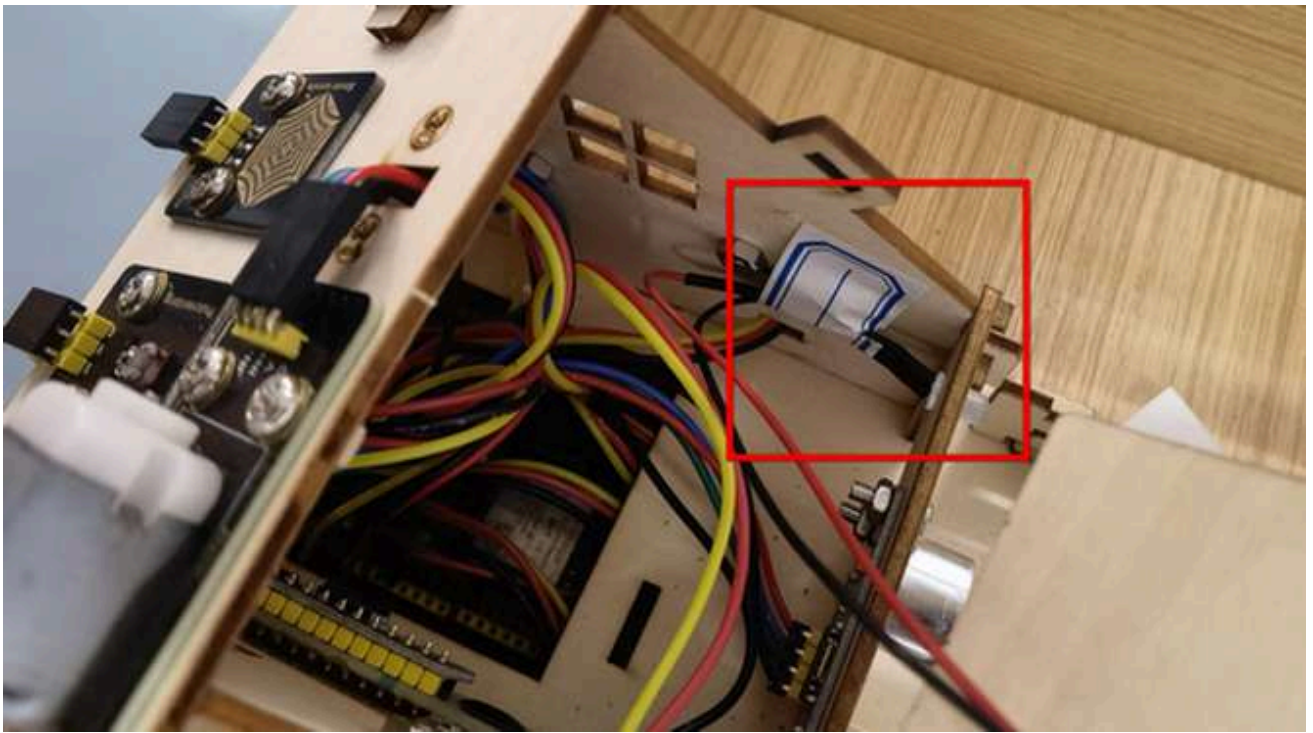
Folosiți un autocolant pentru a-și fixa firele pe perete



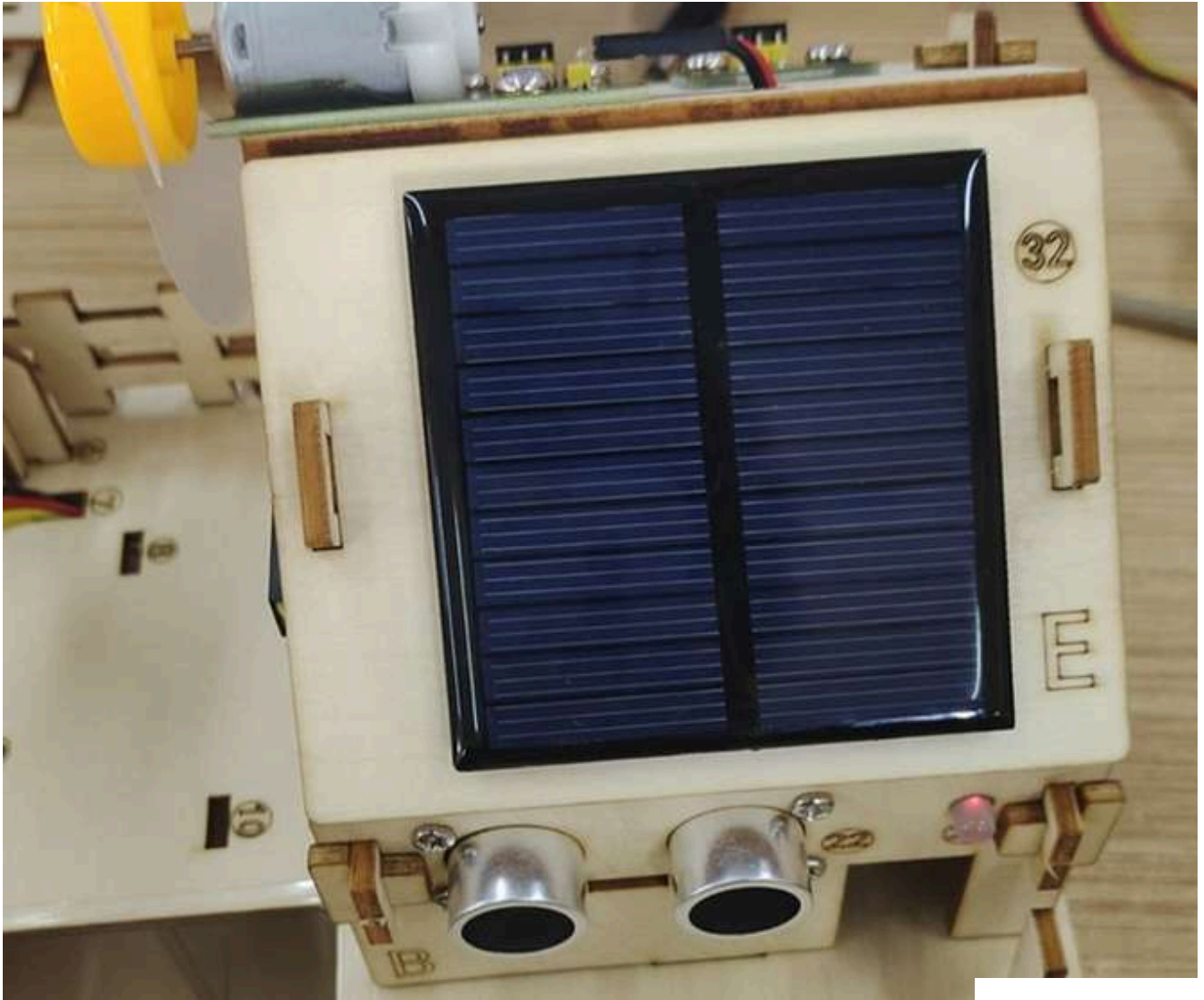
18.7

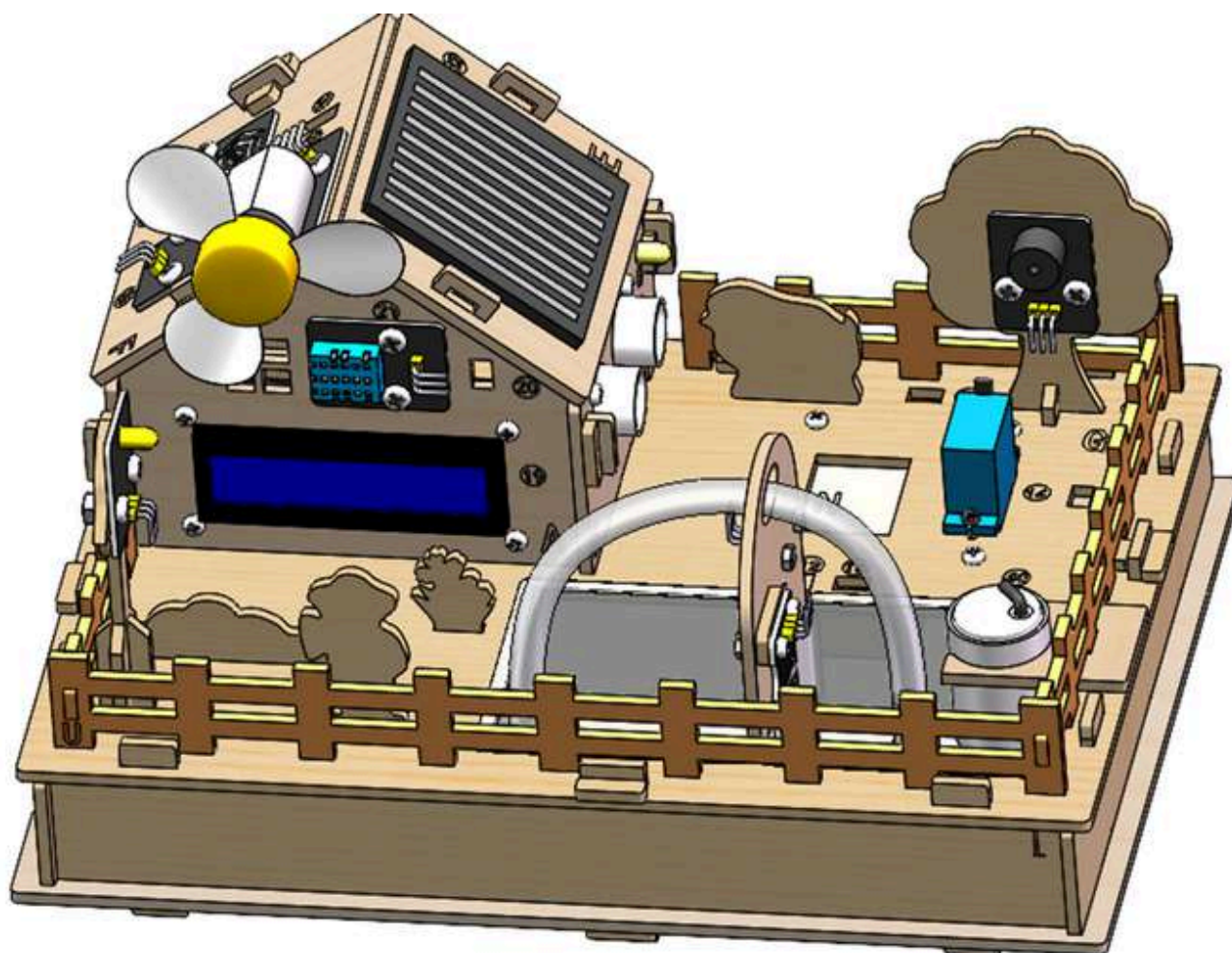


18.8



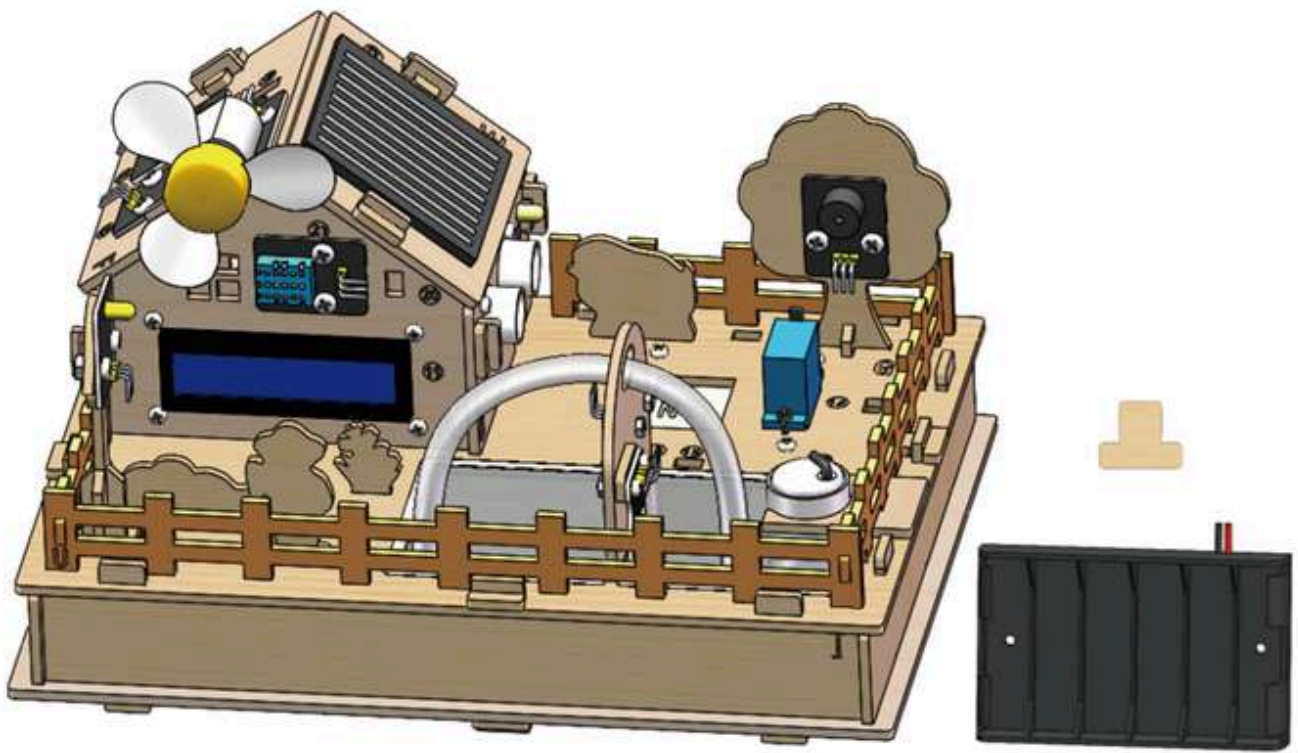
18.9





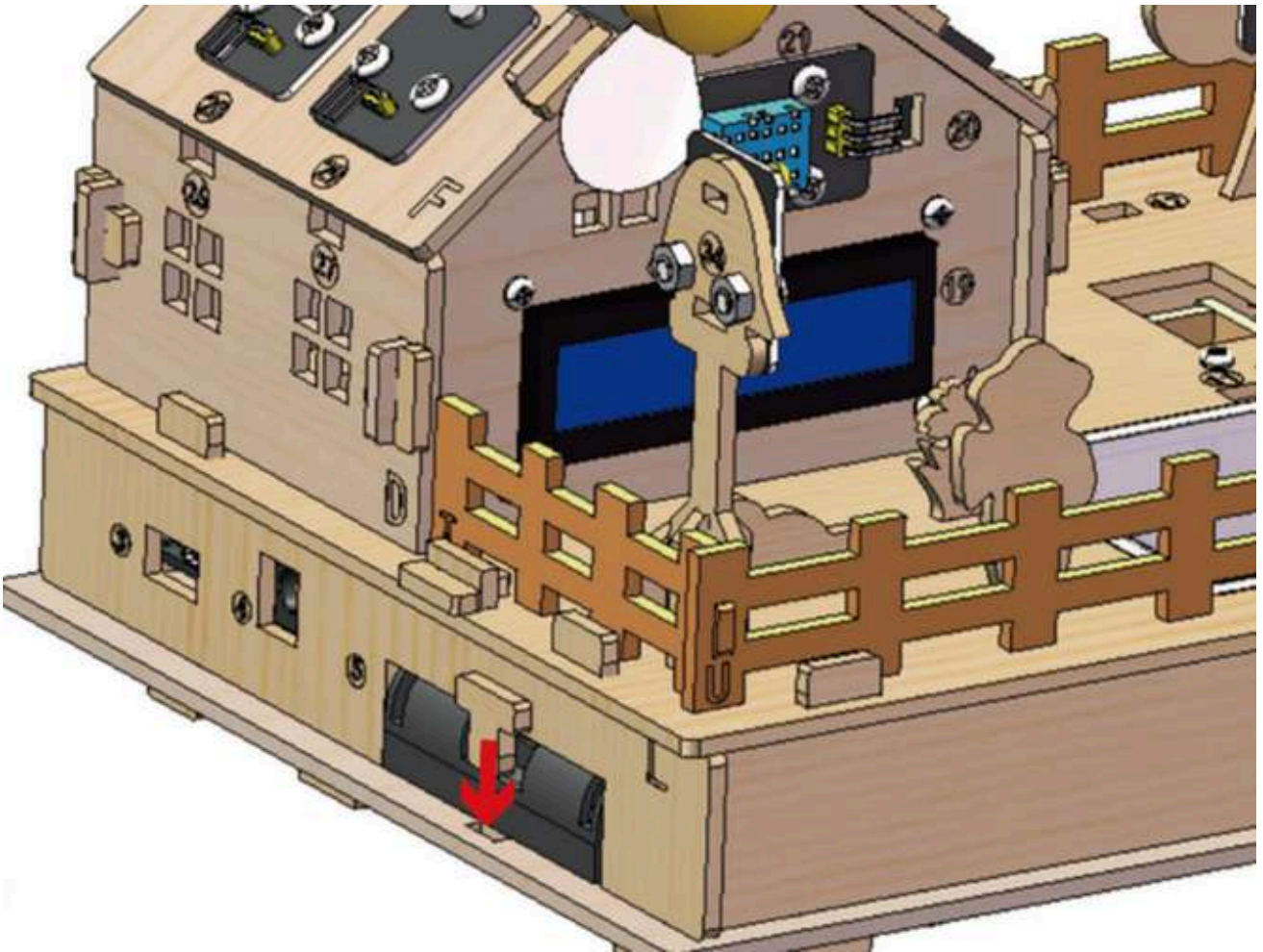
Pasul 19 Instalați carcasa bateriei

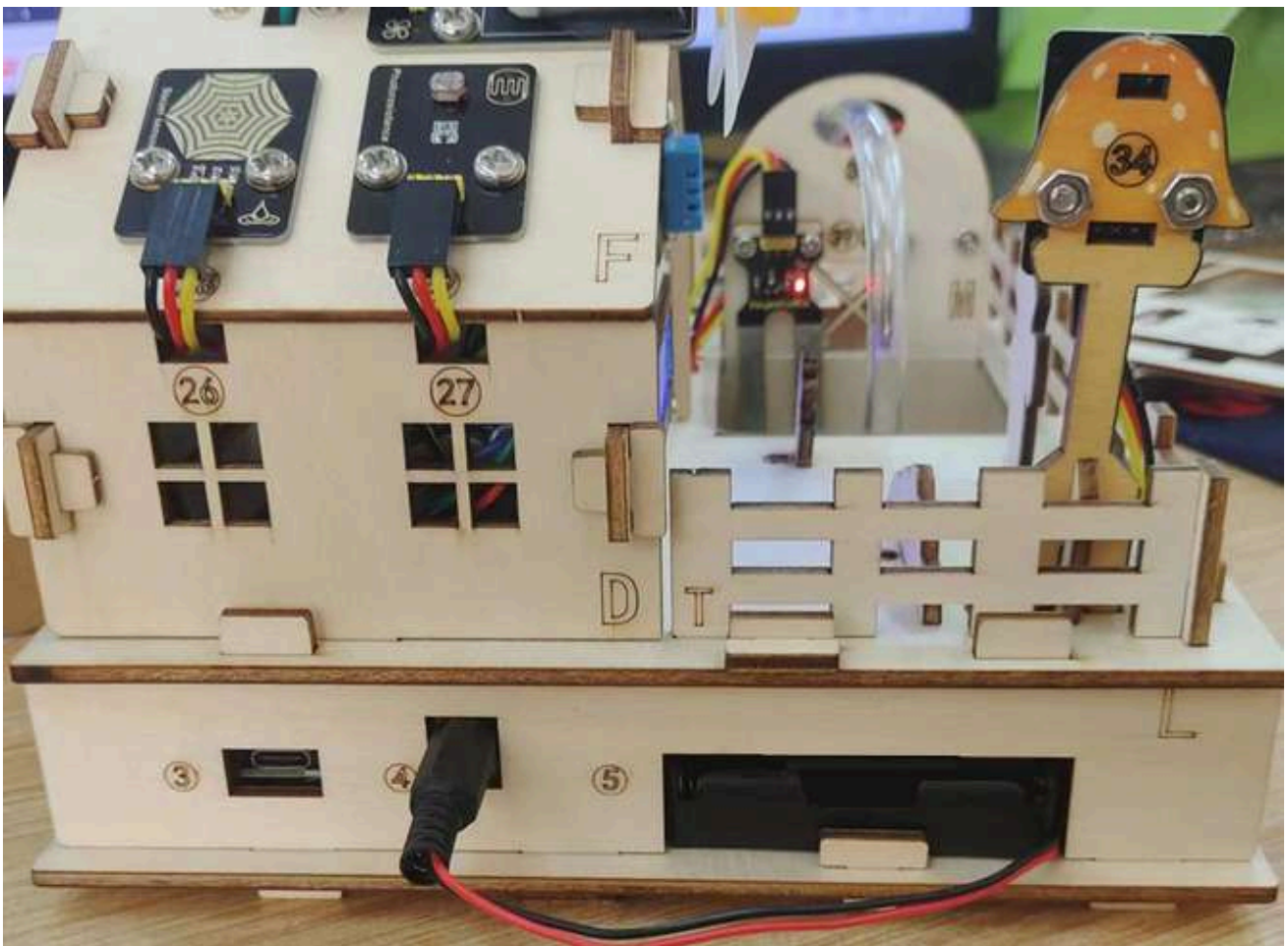
19.1



19.2

Instalați 6 baterii AA (nu sunt incluse în kit)





4. Proiecte

4.1 Proiect: Sistem de iluminat

Să începem primul nostru proiect, sistemul de iluminat.

Aprinderea unui LED este una dintre cele mai fundamentale practici Arduino.

Această lecție de pornire este concepută pentru începători pentru a înțelege hardware-ul și programare software pe placa de dezvoltare ESP32 și pentru a stăpâni cunoștințe de circuit și programare.

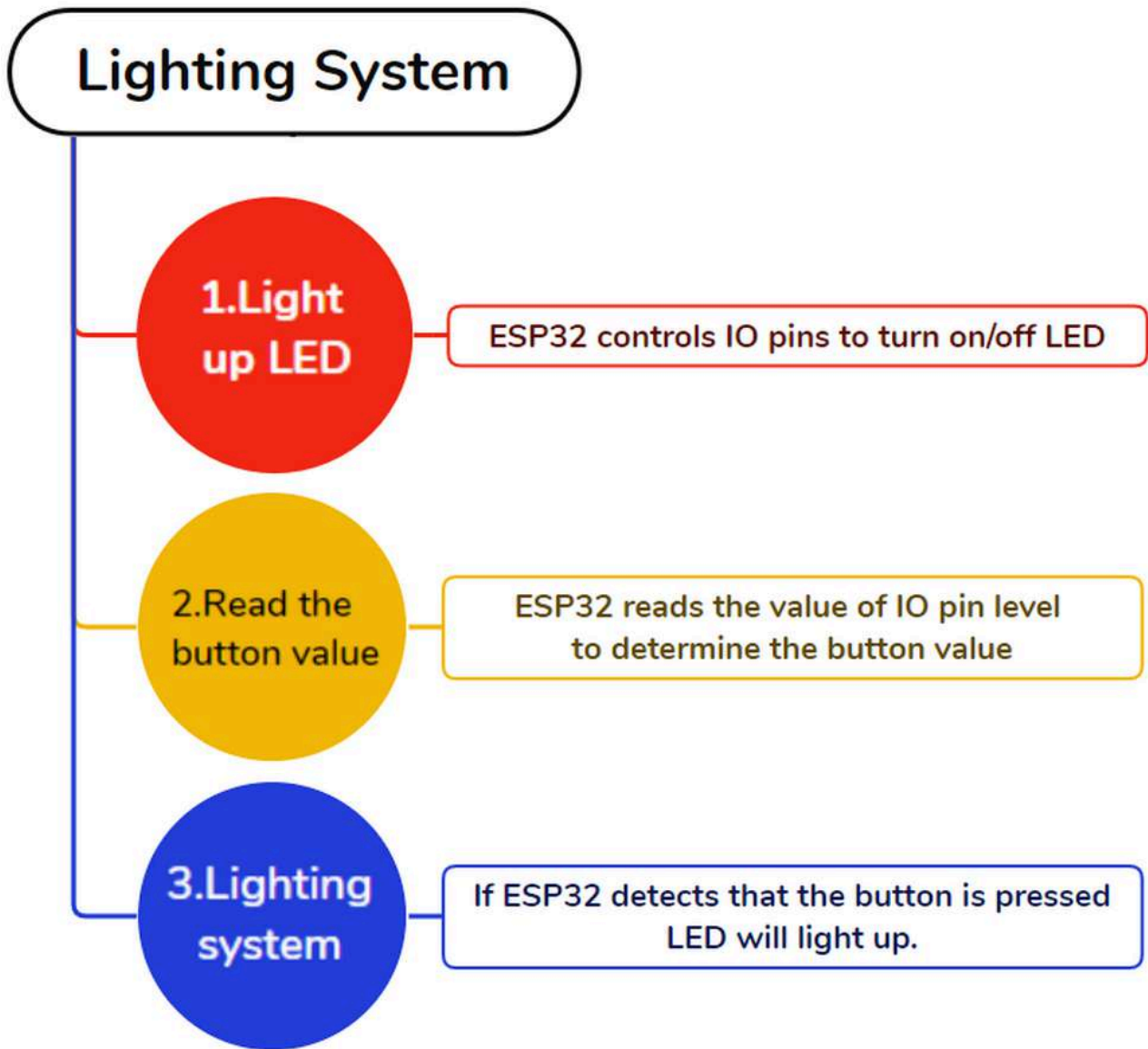


Prin urmare, îndrumarea noastră tutorial este simplă. Și acest proiect intrigant poate fi aplicat în scenarii reale acasă sau la birou.

În acest proiect, veți fi învățat conexiunile de bază și setările plăcii de dezvoltare ESP32 în programarea Arduino. În plus, vă vor fi prezentate și unele funcții, cum ar fi aprinderea/stingerea unui LED prin nivelul de ieșire al unui pin digital sau printr-un nasture.

Una peste alta, acesta este un tutorial entry-level pentru a pune bazele practice
Arduino.

4.1.1 Diagrama de flux



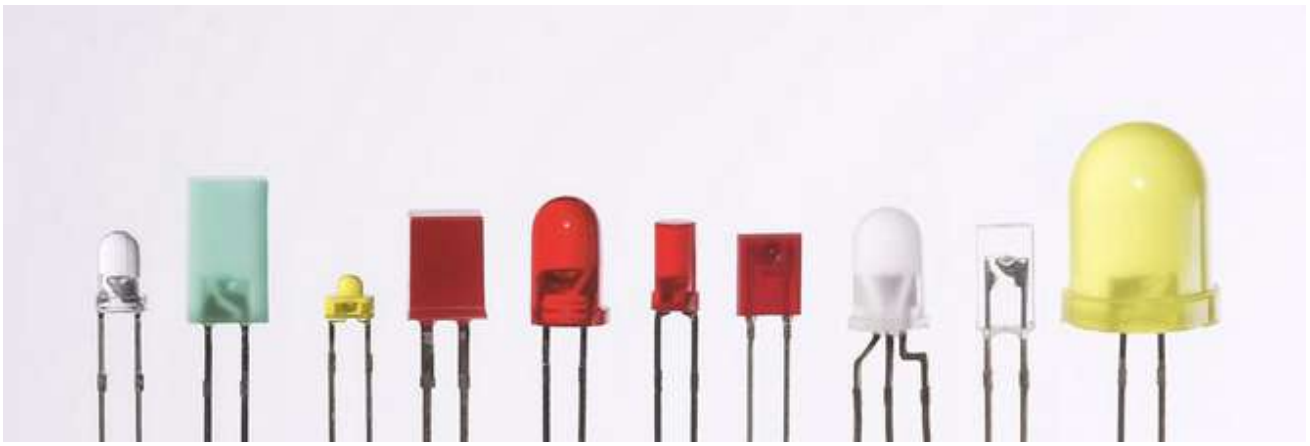
4.1.2 Aprindeți un LED

Descriere:

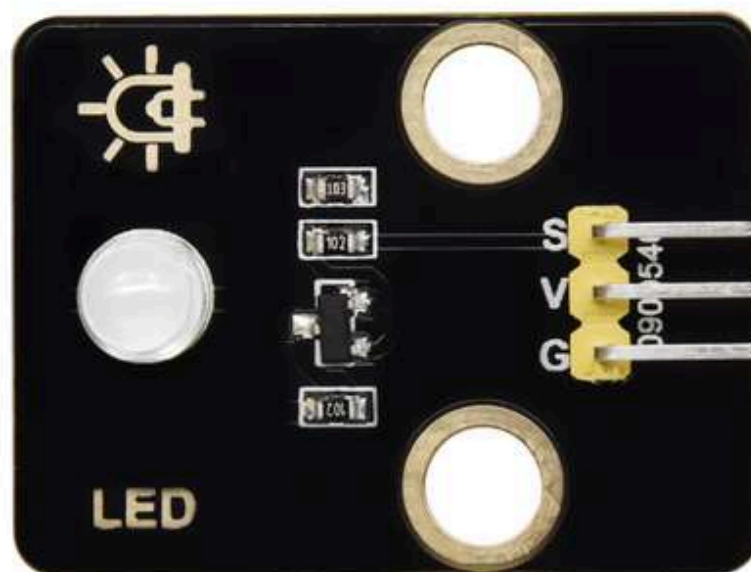
LED-ul, prescurtare de la Light Emitting Diode, este un semiconductor în stare solidă care transformă energia electrică în lumină vizibilă, deci se mai numește și iluminare în stare solidă.

Când curentul trece printr-un LED, acesta se aprinde.


Diverse LED-uri:

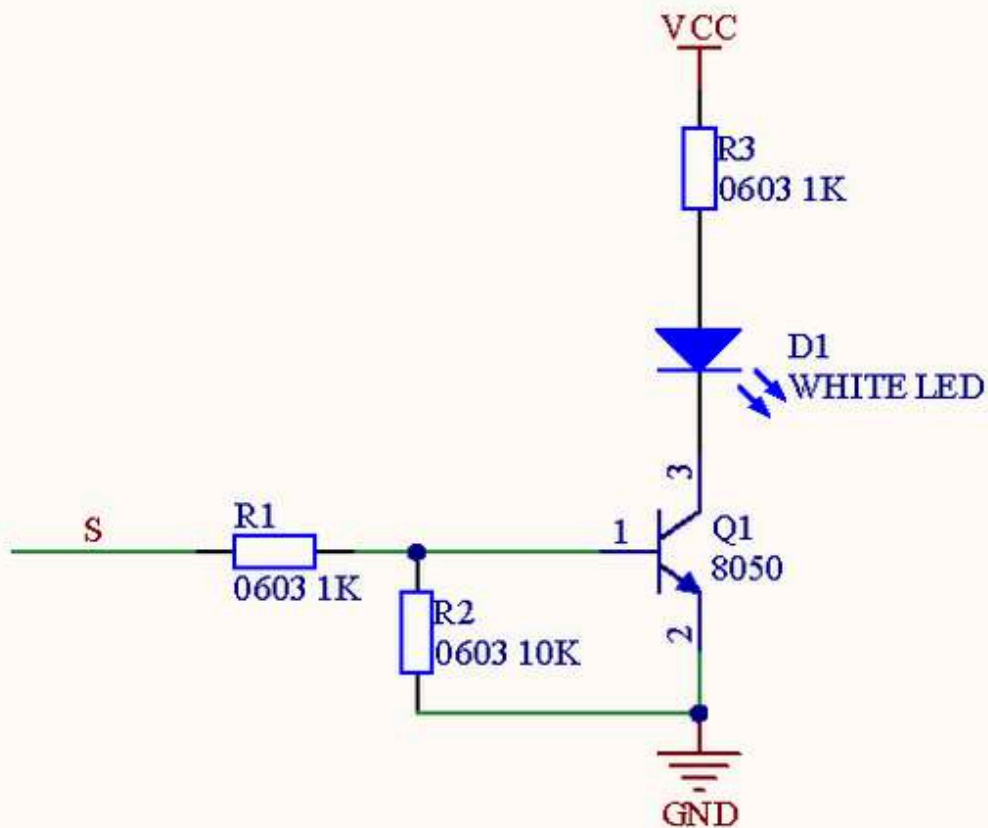


Modulul LED este un dispozitiv de ieșire, a cărui luminozitate și clipiri pot fi controlat. Pentru modul de utilizare, trebuie doar să-l conectați direct la pini de ieșire digitală pe placa de dezvoltare.



Principiul de funcționare:

Când S este la un nivel ridicat, trioda Q1 este în conducție și tensiunea VCC  [Ultima](#) ▼ pentru a-l aprinde.



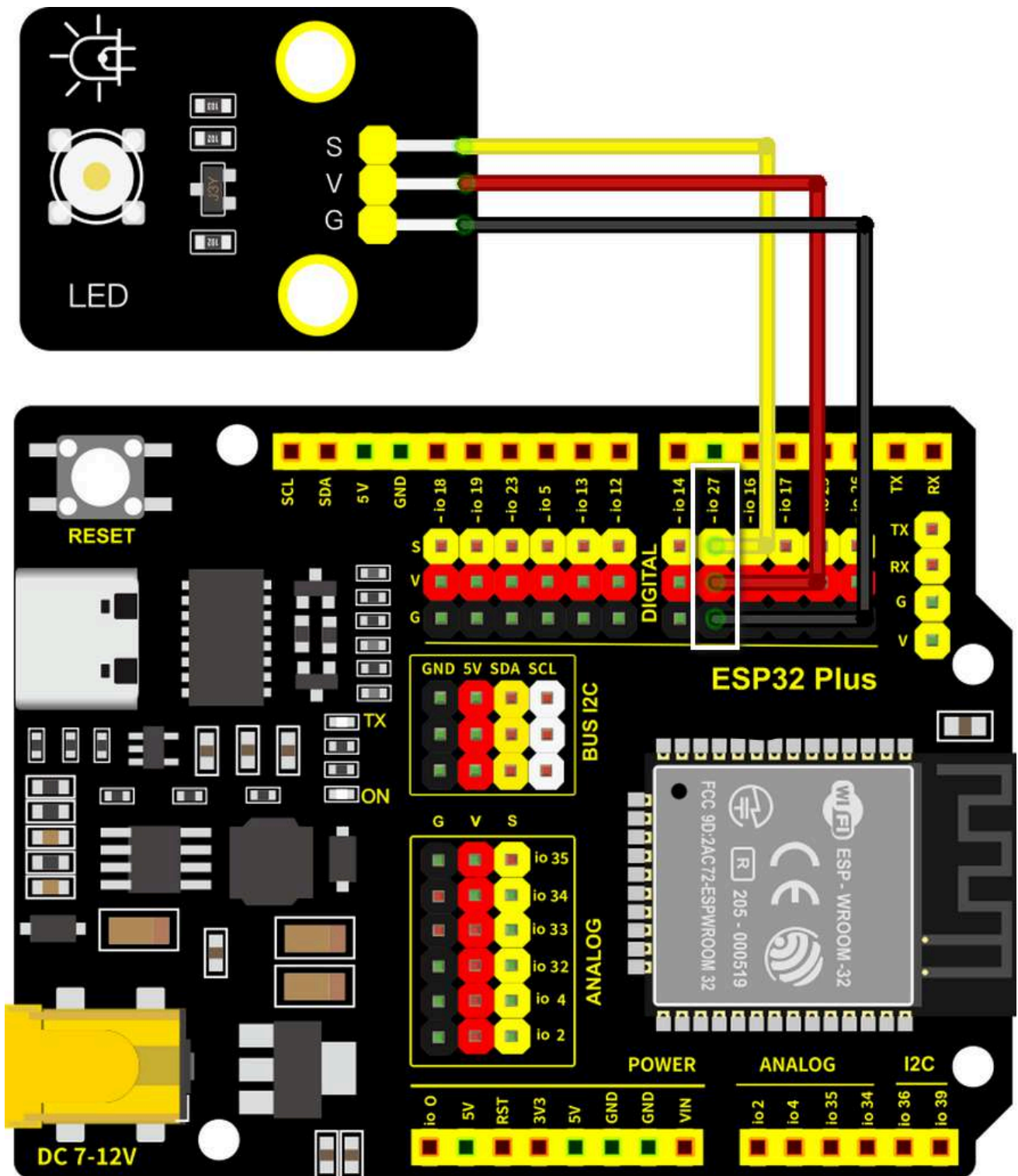
Parametrii:

- Tensiune: 3 ~ 5V
- Curent: $\leq 1,5$ mA
- Putere: 0.07W

Schema:

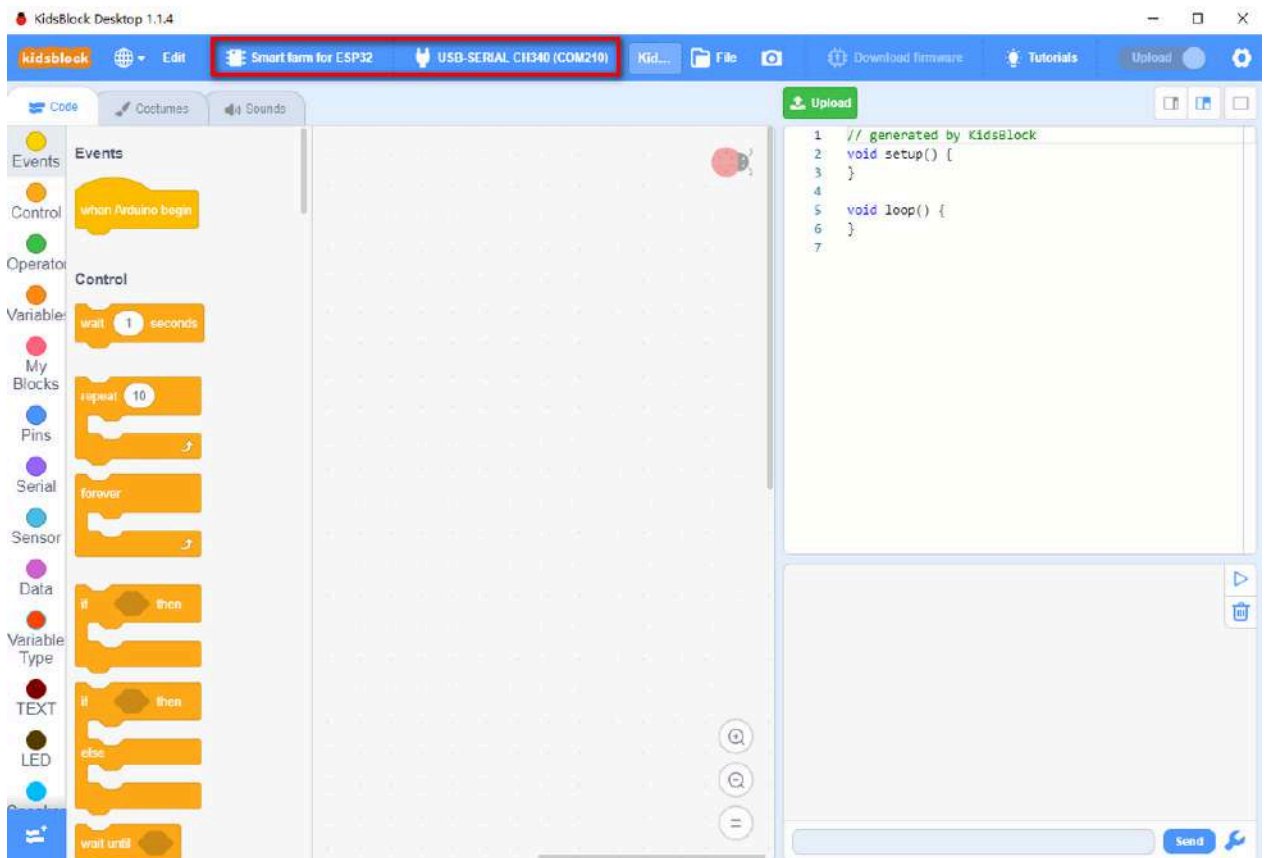
Conectați modulul LED la io27.

Atenție: Conectați galben la S (Semnal), roșu la V (Alimentare) și negru la GND. Nu le inversați!



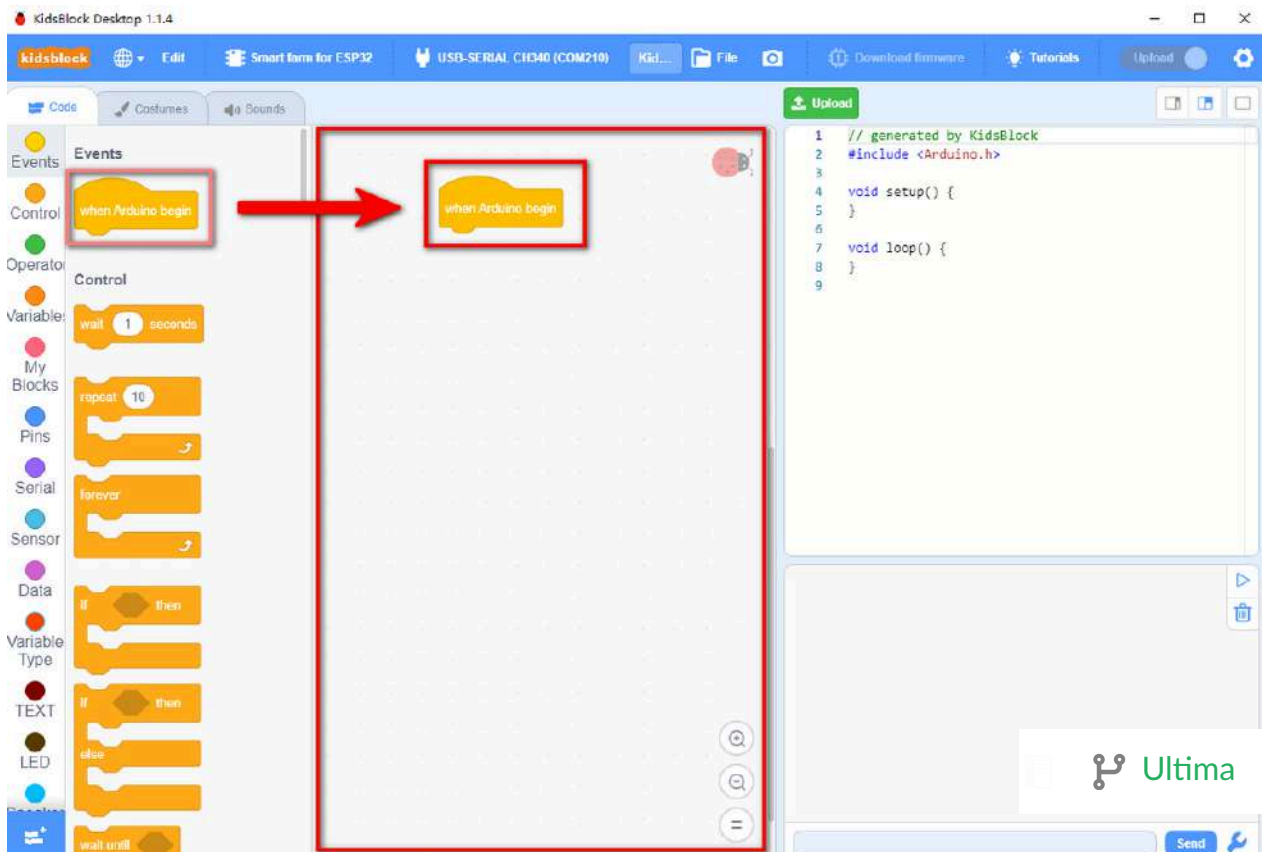
Cod de testare:

- Deschideți Kidsblock și alegeți dispozitivul și portul corect.

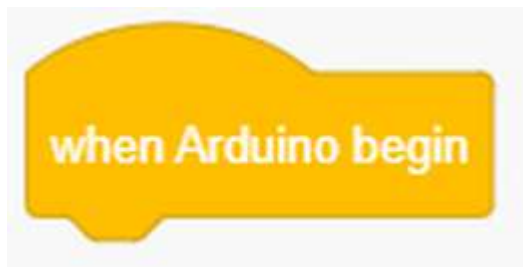



- Trageți  din zona de  editare a codului.

Blocuri de cod execută numai atunci când se află în această zonă.




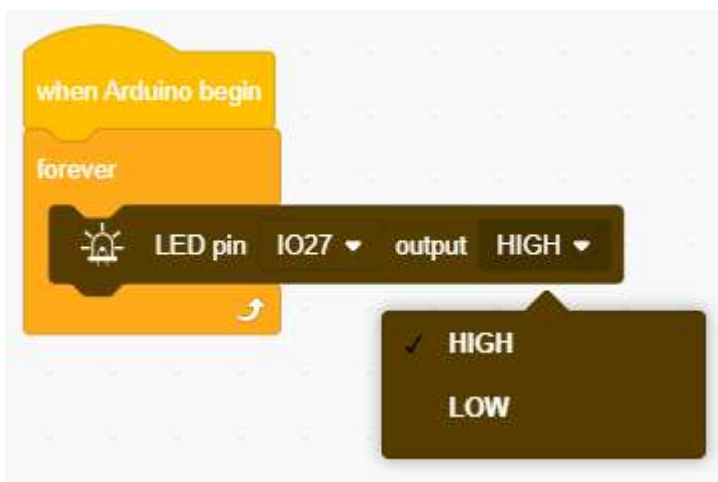
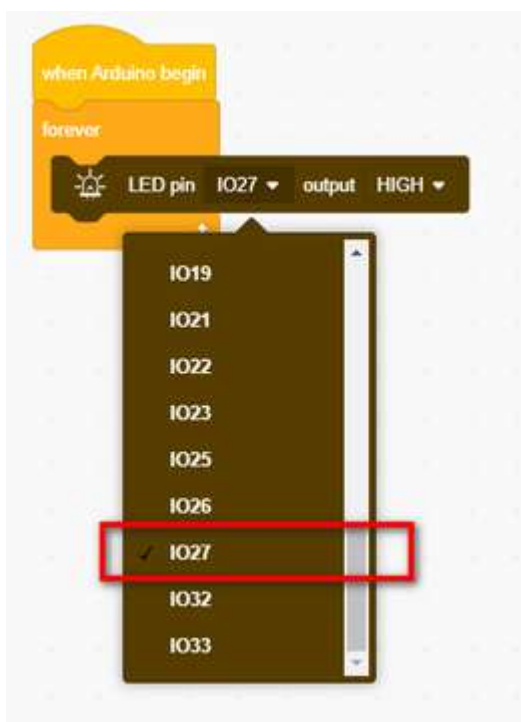
- Cu acest bloc, la pornirea plăcii de dezvoltare, codul va rula.



- În  Control, trageți "pentru totdeauna" și lipiți-l sub precedentul bloca. Blocul "pentru totdeauna" indică o buclă.



- Drag an "LED pin output" block from  LED and paste it in "forever". Set the pin to IO27 and output level to HIGH, so that the LED pin will continue to output high level.



- Add a 1s delay. Duplicate the "LED pin output" block but set the output to LOW and add a delay. Then LED will light up and go off in circulation.



Rezultatul testului:

LED-ul clipește pe secundă, deoarece io27 de pe placa ESP32 iese sus și jos nivel alternativ în fiecare secundă. În plus, diverse aplicațiile pot fi realizate și prin intermediul unui LED, cum ar fi LED-ul de respirație, apa lumini de curgere și lumină intermitentă a poliției.

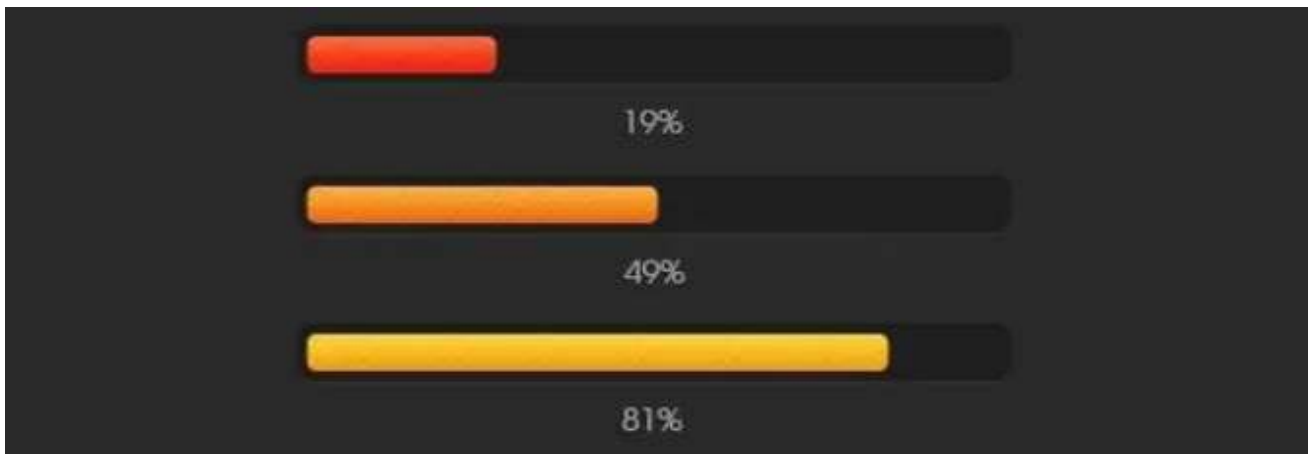
Nivel de putere	Rezultat
ÎNALT	LED aprins
JOS	LED stins

Expansiune: LED de respirație

Descriere:

Interfețe IO ale MCU (arduino UNO, ESP32 și Raspberry Pi Pico) numai semnale digitale (nivel înalt sau scăzut). De exemplu, în (aprinde un LED), ieșirile digitale sunt doar HIGH (3.3V) și SCĂZUT (0V).

Dacă MCU emite un nivel ridicat de 3,3 V sau un nivel scăzut de 0 V, tensiunea ar trebui să fie la 0~3.3V. Astfel, PWM (**Pulse Width Modulation**) este necesar pentru a scoate o valoare de tensiune diferită, care se numește "analog ieșire".

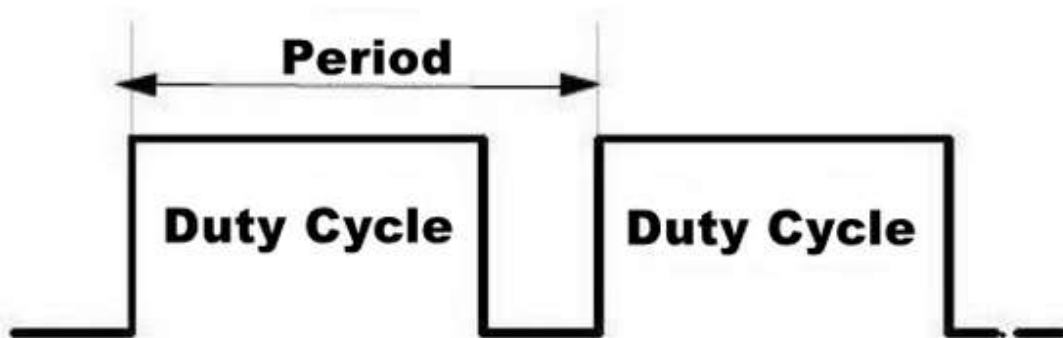


Cunoaștere:

Ce este PWM?

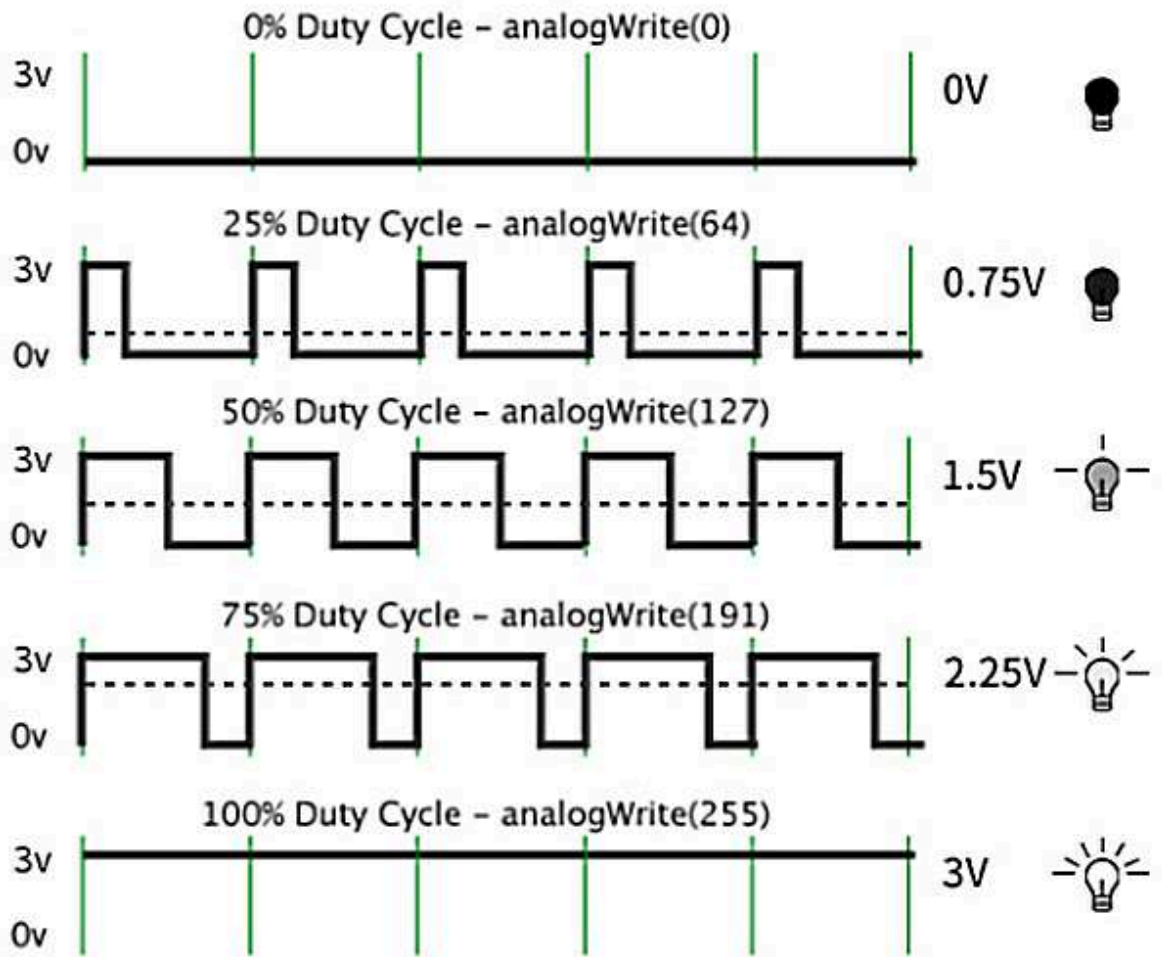
PWM conține trei elemente: frecvență (Hz), perioadă, ciclu de lucru (%).

- **Frecvența PWM (f):** timpii de schimbare a semnalului de la înalt la scăzut și reveniți la sus într-o secundă. În general, Frecvență este numărul de puncte PWM într-o secundă.
- **Perioada PWM (T):** Perioada = $1 / \text{Frecvență}$ ($T=1/f$, iar 1 înseamnă 1 al doilea). De exemplu: $f = 50\text{Hz}$, deci $T = 20\text{ms}$, ceea ce implică că există De 50 de ori de Punct pe secundă.
- **PWM Duty Cycle:** raportul de timp dintre HIGH și întreaga perioadă. Dacă Perioada = 10 ms și 8 ms este timpul de lățime a impulsului, nivelul scăzut ocupă 2 ms, deci ciclul de lucru = $8/(8+2) = 80\%$.

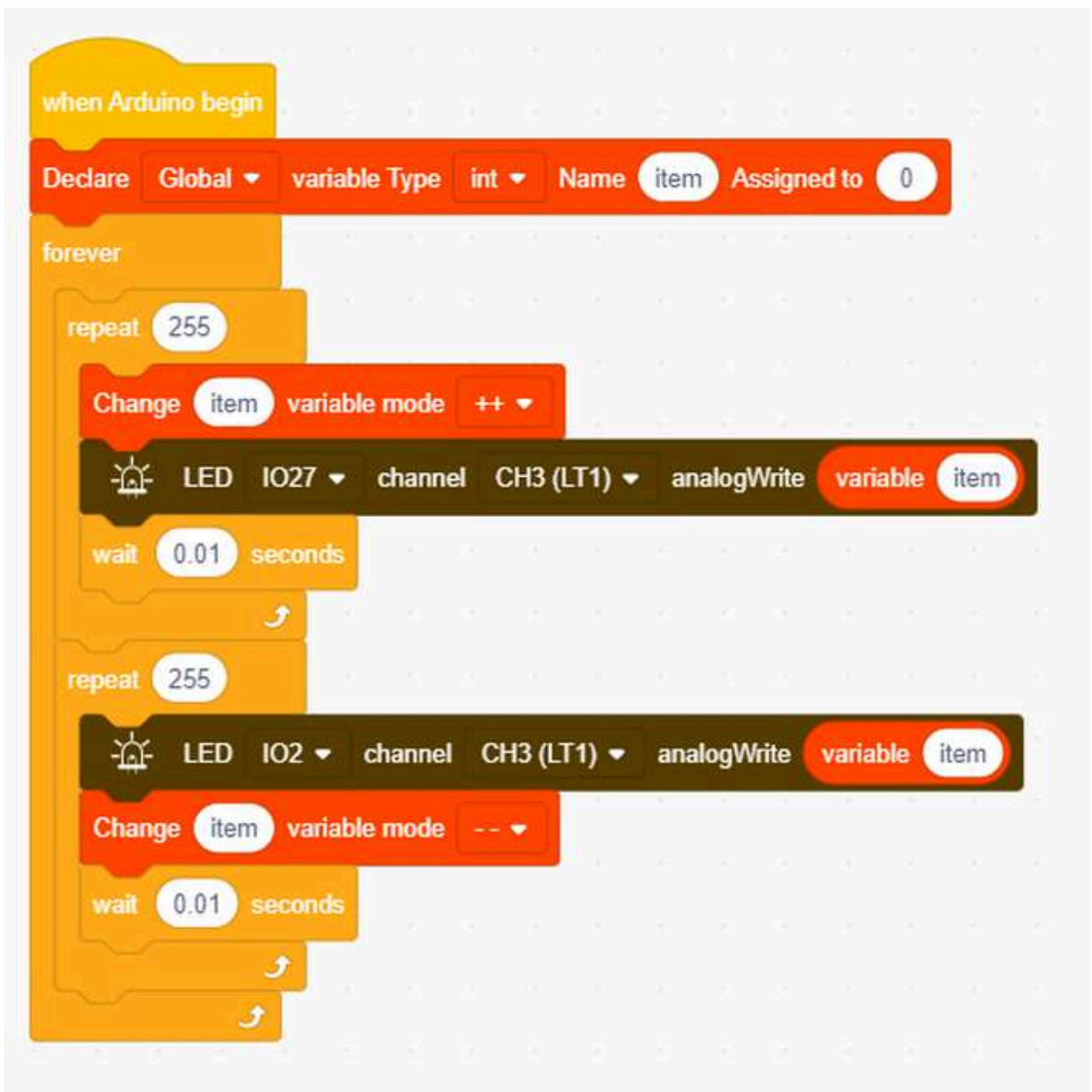


Concluzie: La o frecvență de semnal adecvată, PWM schimbă efectiv tensiune de ieșire prin modificarea ciclului de lucru într-o singură perioadă. În câmpie Română, într-un anumit timp, cu atât portul IO este mai înalt ieșiri, cu cât este mai mare valoarea PWM și LED-ul mai deschis va fi.

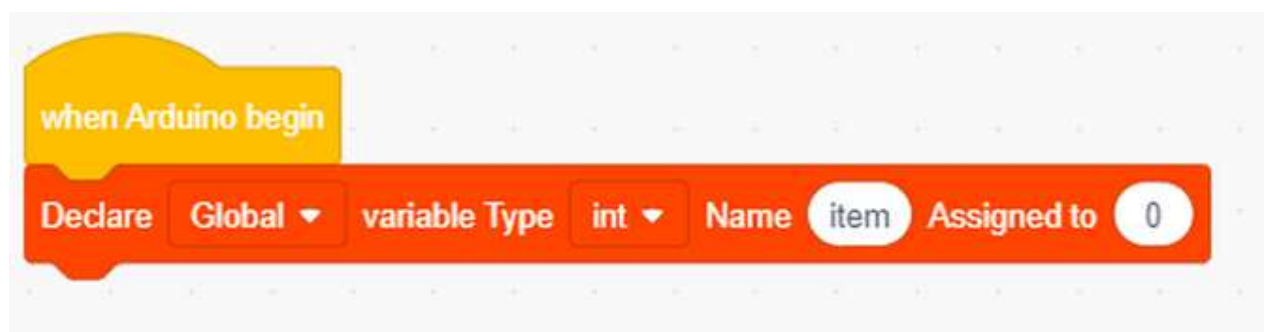
Pulse Width Modulation



Cod de testare:



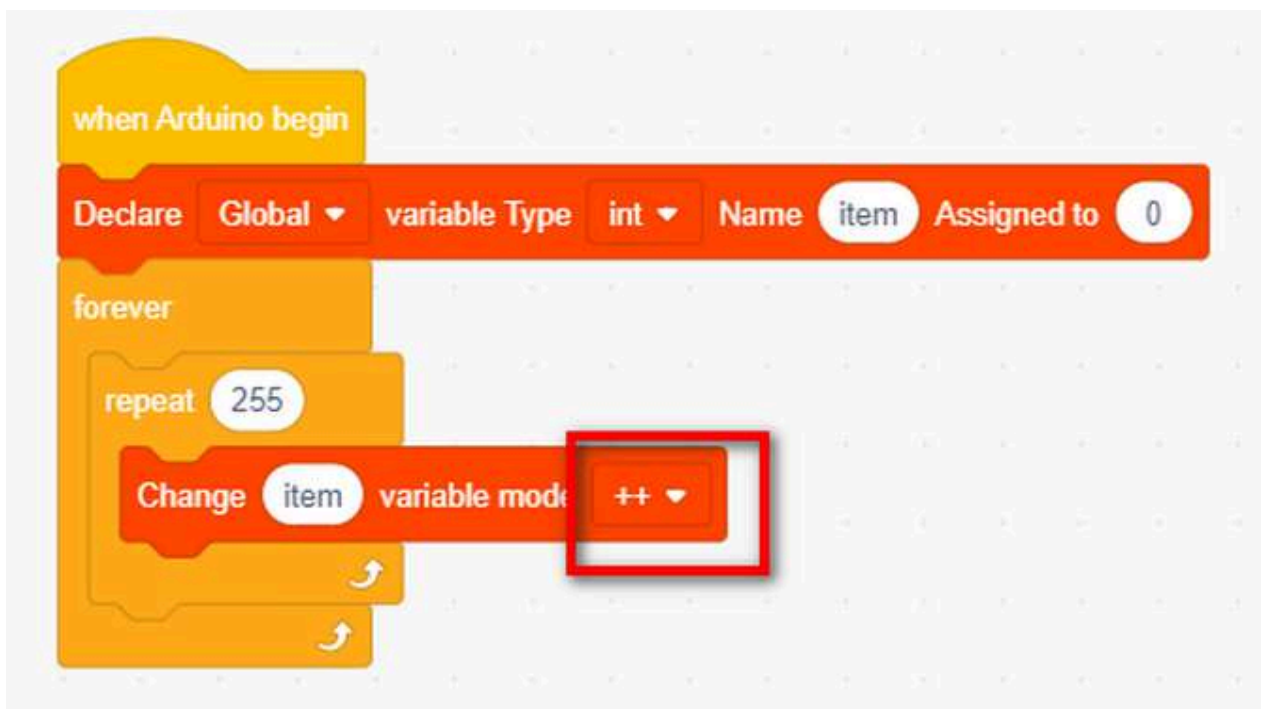
- Definiți un element variabil și atribuiți-l la 0.




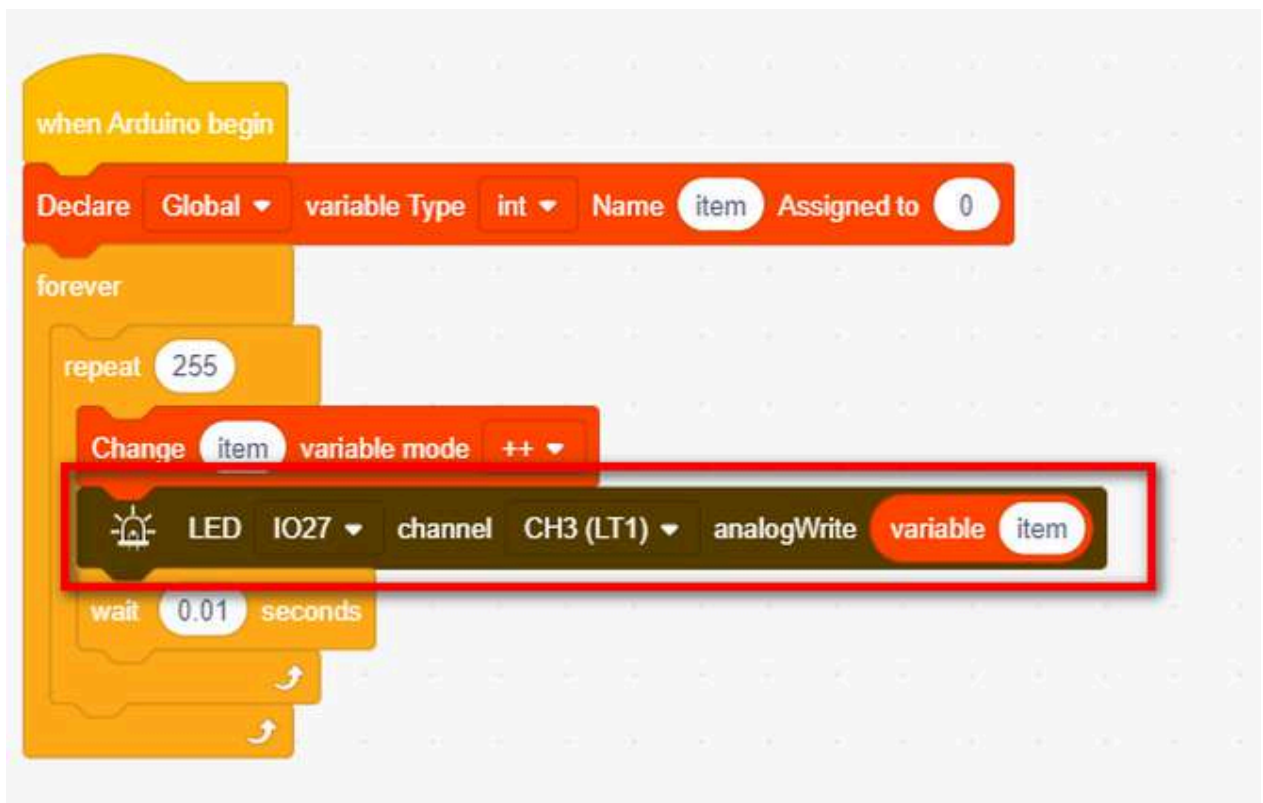
- Trageți un bloc "pentru totdeauna" și lipiți un bloc "repetare" în el. Apus repetați timpii la 255.



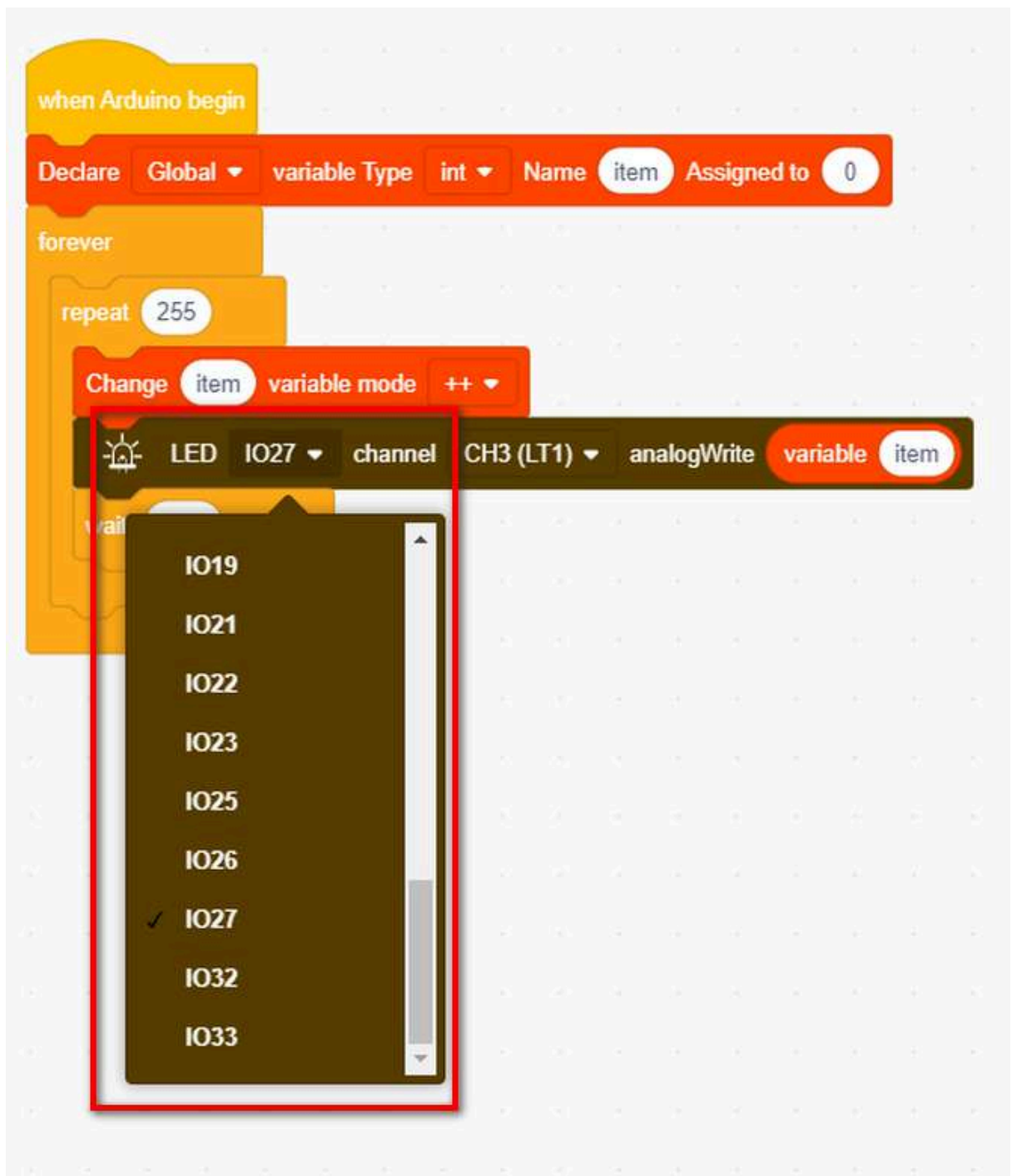
- Trageți un bloc "mod variabil" în "repetare" și setați modul la "++", ceea ce înseamnă că **obiectul** va crește cu 1 după fiecare execuție.



- Găsiți blocul pentru a seta PWM care este conținut așa  cum se arată de mai jos, deci trebuie doar să setați pinul corespunzător și valoarea analogică la PWM de ieșire.



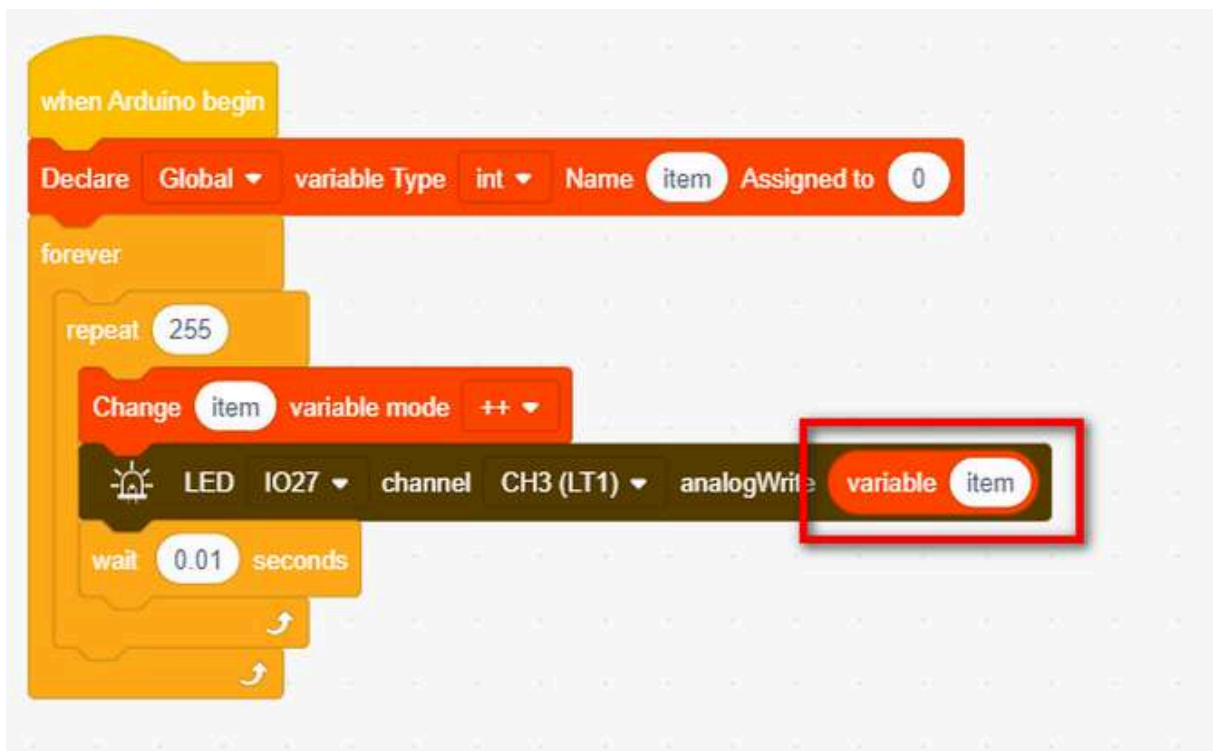
- Setăți pinul LED:



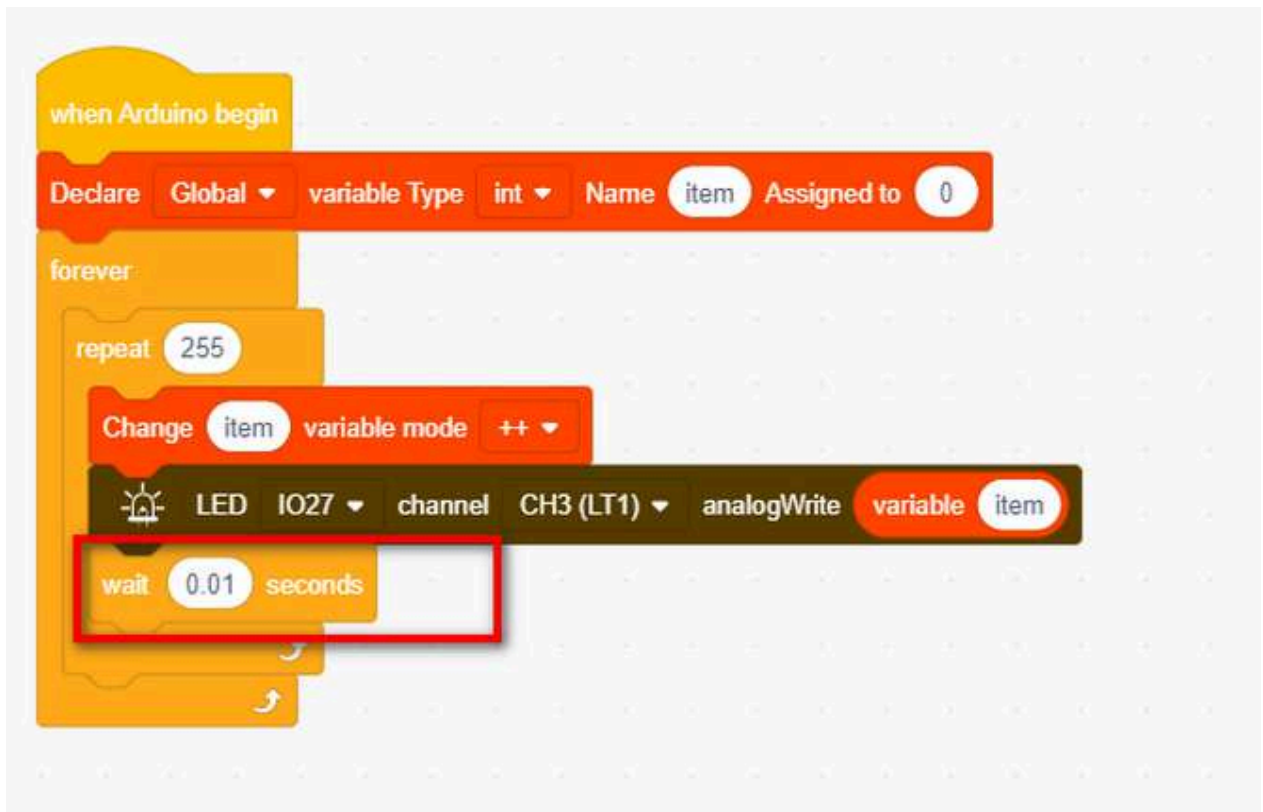
- Setați canalul: (16 canale în total: inclusiv 0~15)



- Setăți valoarea de ieșire PWM la element, care va adăuga automat 1 de la 0 la 255. Ieșirea PWM este de 0 ~ 255, deci setăm timpii de repetare la 255.



- Adăugați o întârziere la 0.01s, astfel încât LED-ul să se aprindă treptat, mai degrabă decât dintr-o dată.

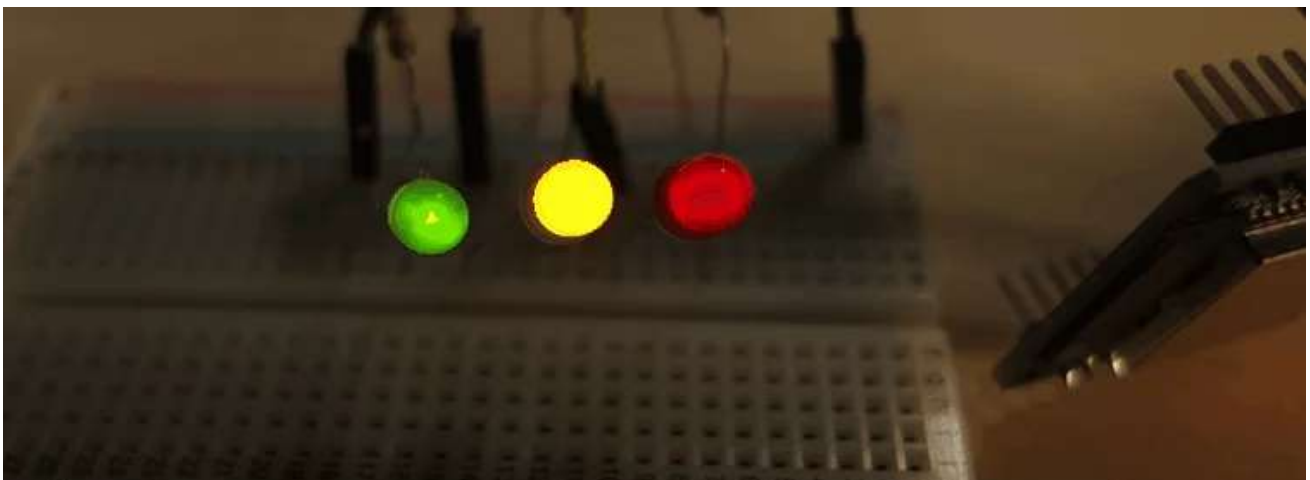


- Duplicați blocul "repetare" după cum urmează, dar setați modul la "--", care scade elementul variabil de fiecare dată. Și LED-ul va se estompează treptat.



Rezultatul testului

LED-ul se aprinde și se estompează treptat; respiră uniform.



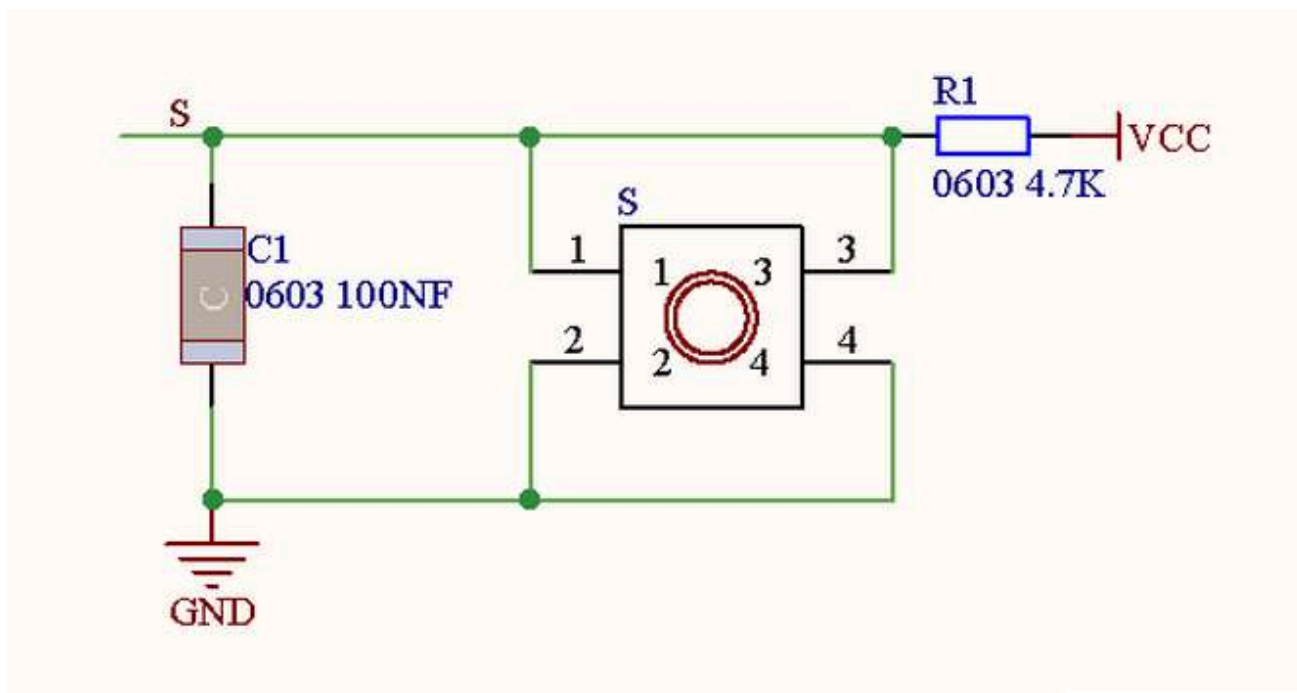
4.1.4 Buton A

Descriere

Modulul buton este un dispozitiv de introdus. MCU își citește nivelul de putere la detectați dacă butonul este apăsat.



Diagramă schematică:



Parametrii:

- Tensiune: 3 ~ 5V
 - Curent: $\leq 1.1\text{mA}$
 - Putere: $\leq 5.5\text{mW}$
-

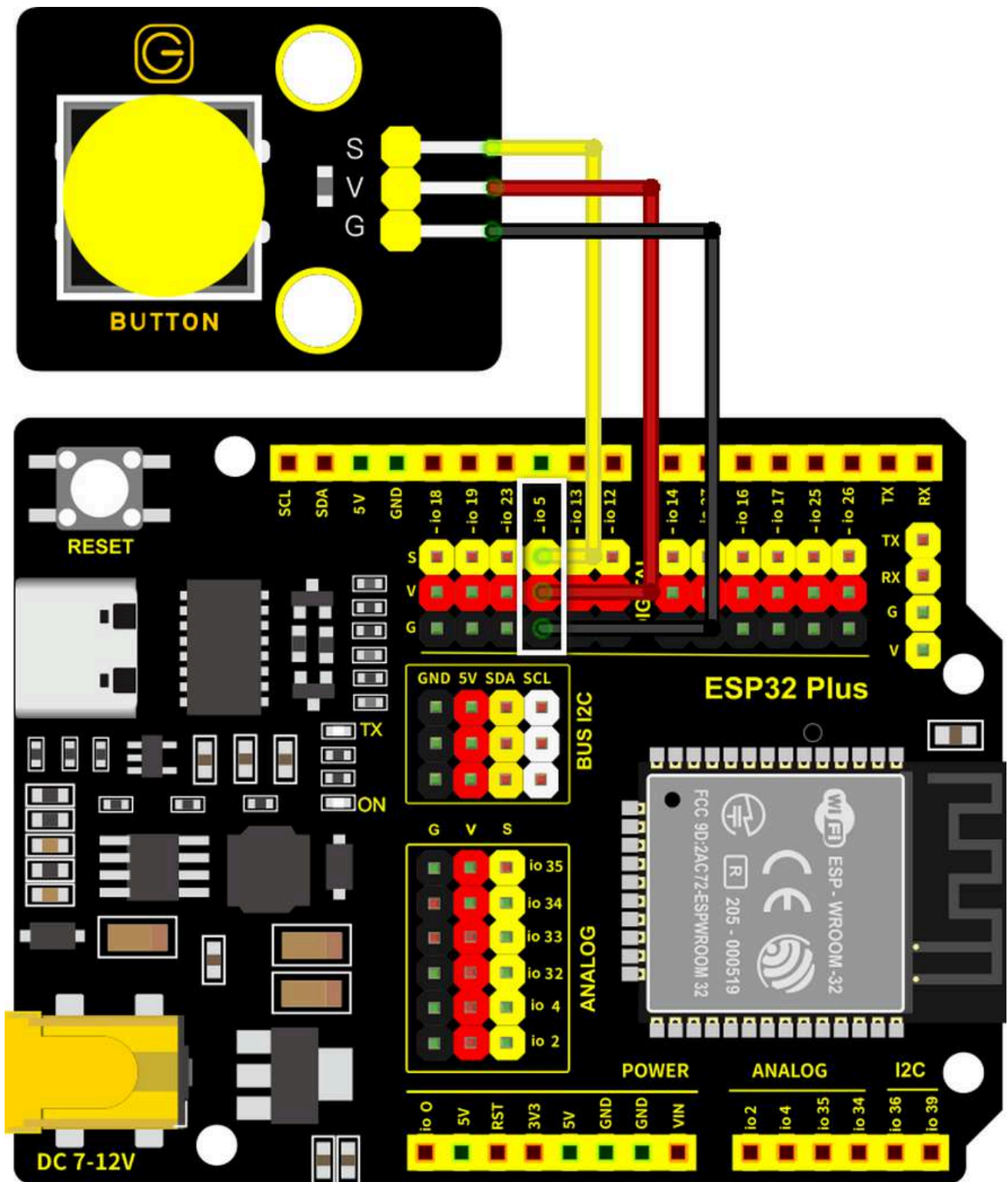
Principiul modului buton este un circuit controlat de acesta nasture.

- **Când butonul este apăsat**, circuitul este în stare închisă, astfel încât acel curent trece prin buton către GND, ceea ce face ca Pin de intrare digital pentru a detecta un nivel scăzut.
 - **Când butonul este eliberat**, circuitul este tăiat și nivelul pinului crește datorită unui rezistor de tragere, ceea ce face ca pinul digital să detecteze un nivel ridicat.
-

Schema:

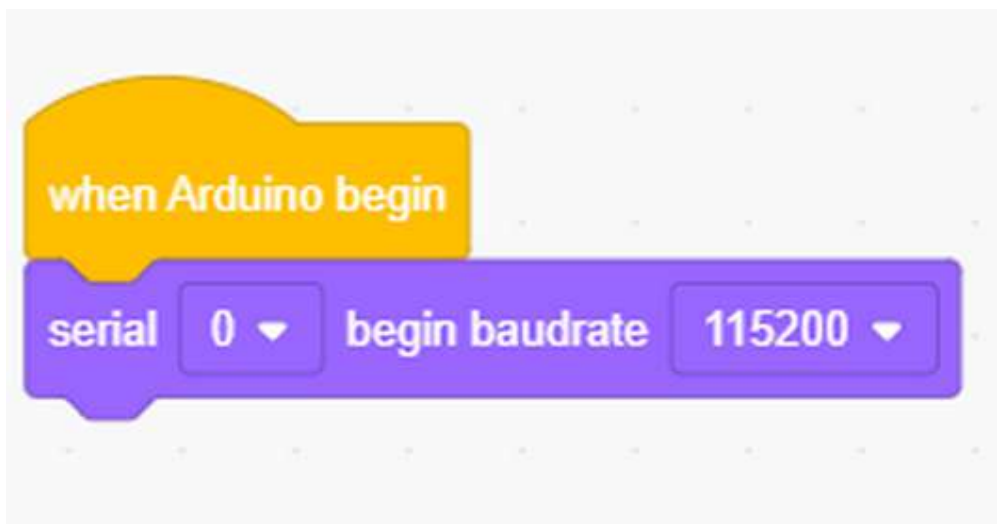
Conectați modulul buton la io5

Atenție: Conectați galben la S (Semnal), roșu la V (Alimentare) și negru la GND. Nu le inversați!

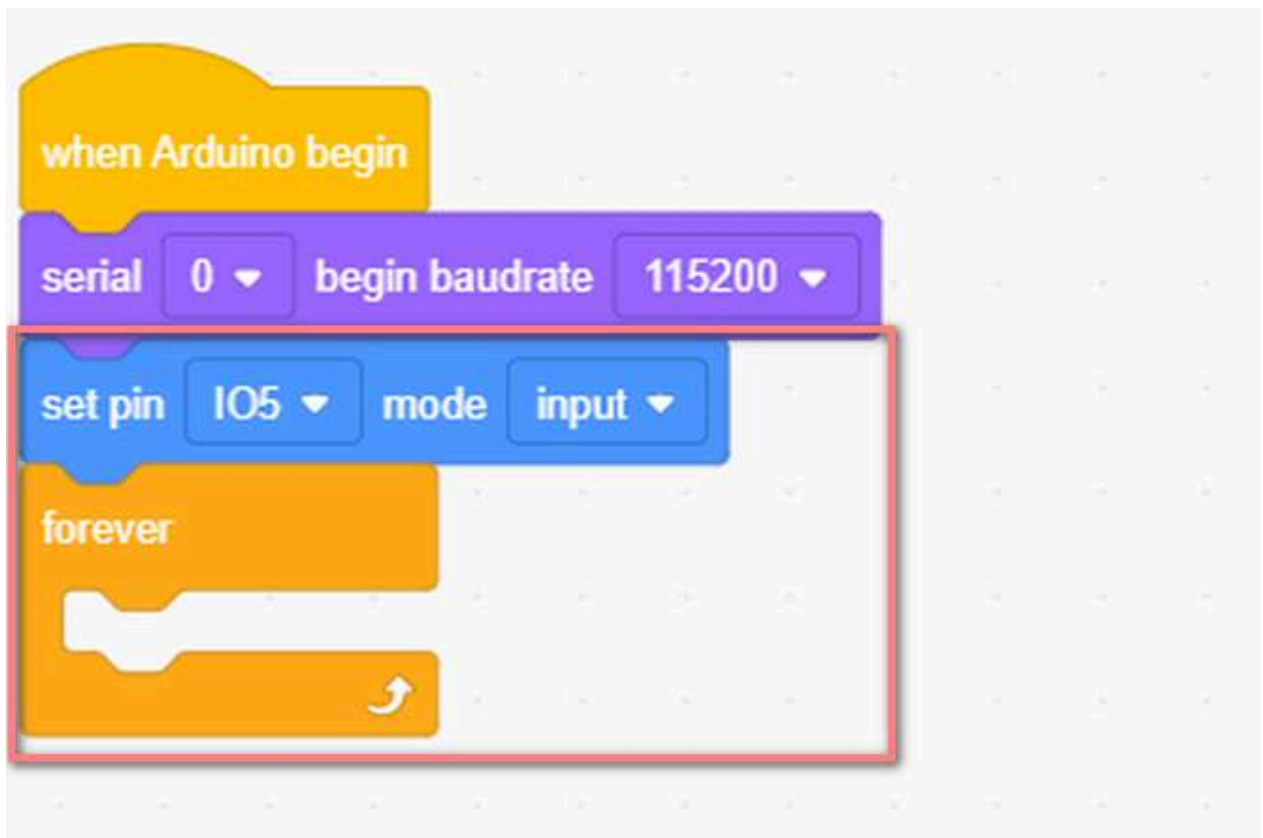


Cod de testare

- Inițializați portul serial în primul rând și setați rata de transmisie la 115200.



- Setezi pinul la IO5 și modul la intrare. Ceea ce urmează este un "pentru totdeauna" bloca.



- Citești nivelul de putere al pinului digital 5. Dacă este 1, imprimați 1. Sau altfel, imprimare 0.

```
if read digital pin IO5 then
  serial 0 print The current status of the button is : 1 warp
else
  serial 0 print The current status of the button is : 0 warp
```

Cod complet:

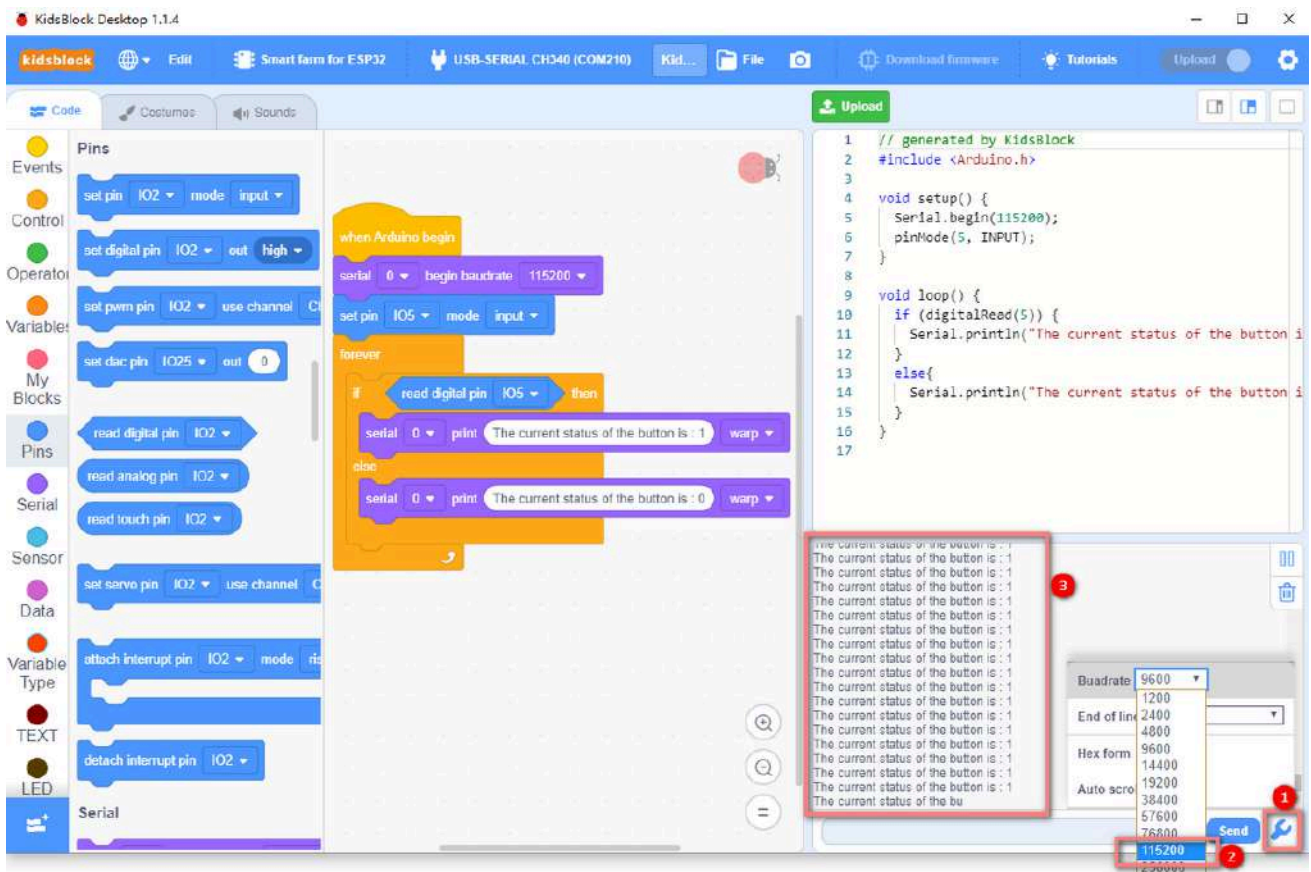
```
when Arduino begin
  serial 0 begin baudrate 115200
  set pin IO5 mode input
  forever
    if read digital pin IO5 then
      serial 0 print The current status of the button is : 1 warp
    else
      serial 0 print The current status of the button is : 0 warp
```

Rezultatul testului

Deschideți monitorul serial și setați rata de transmisie corespunzătoare.

[Ultima](#) ▼

Când butonul este eliberat, valoarea este 1; dacă apăsați butonul, devine 0.



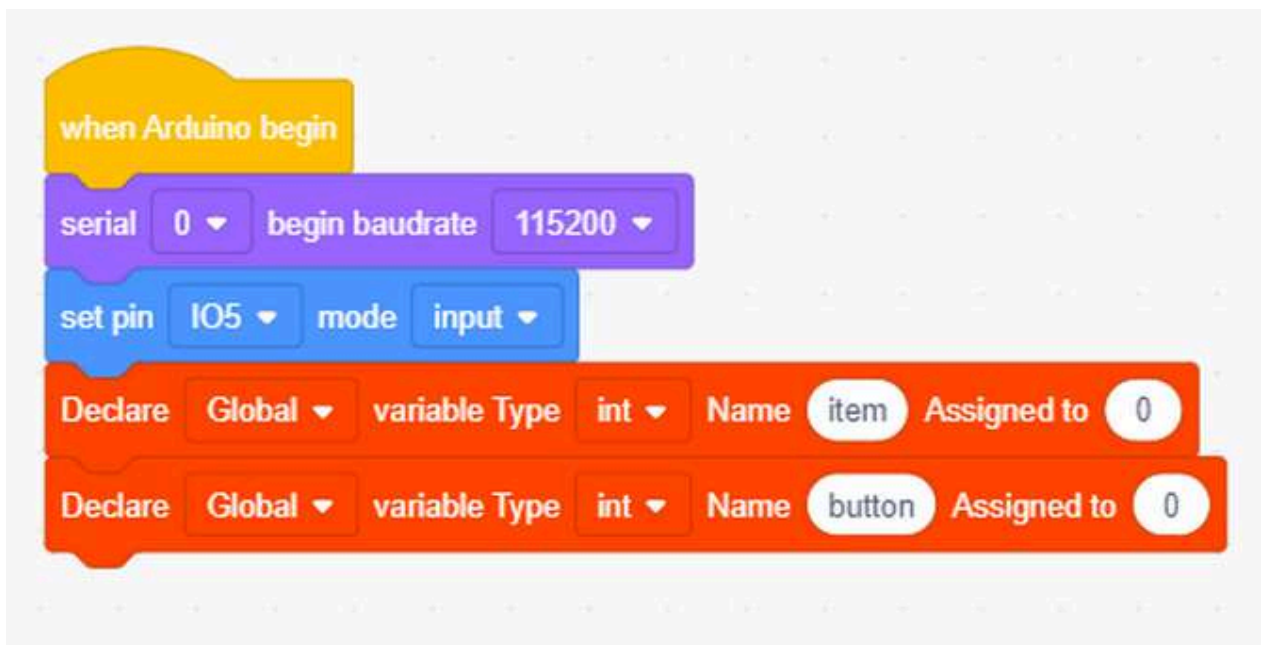
În KidsBlock, putem citi starea pinului de intrare digitală prin programare pentru a detecta dacă butonul este apăsat. Astfel, o mulțime de aplicațiile interactive pot fi realizate printr-un modul de butoane, cum ar fi LED-ul pornit/oprit și reglarea luminozității afișajului.

Expansiune: Buton de blocare automată

Un buton de blocare automată nu va apărea atunci când îl apăsați fără a ține apăsat, și nu apare niciodată decât dacă îl apăsați din nou. Funcționează ca un comutator. Pentru butoanele obișnuite, o astfel de funcție poate fi realizată prin MCU și software.

Cod de testare

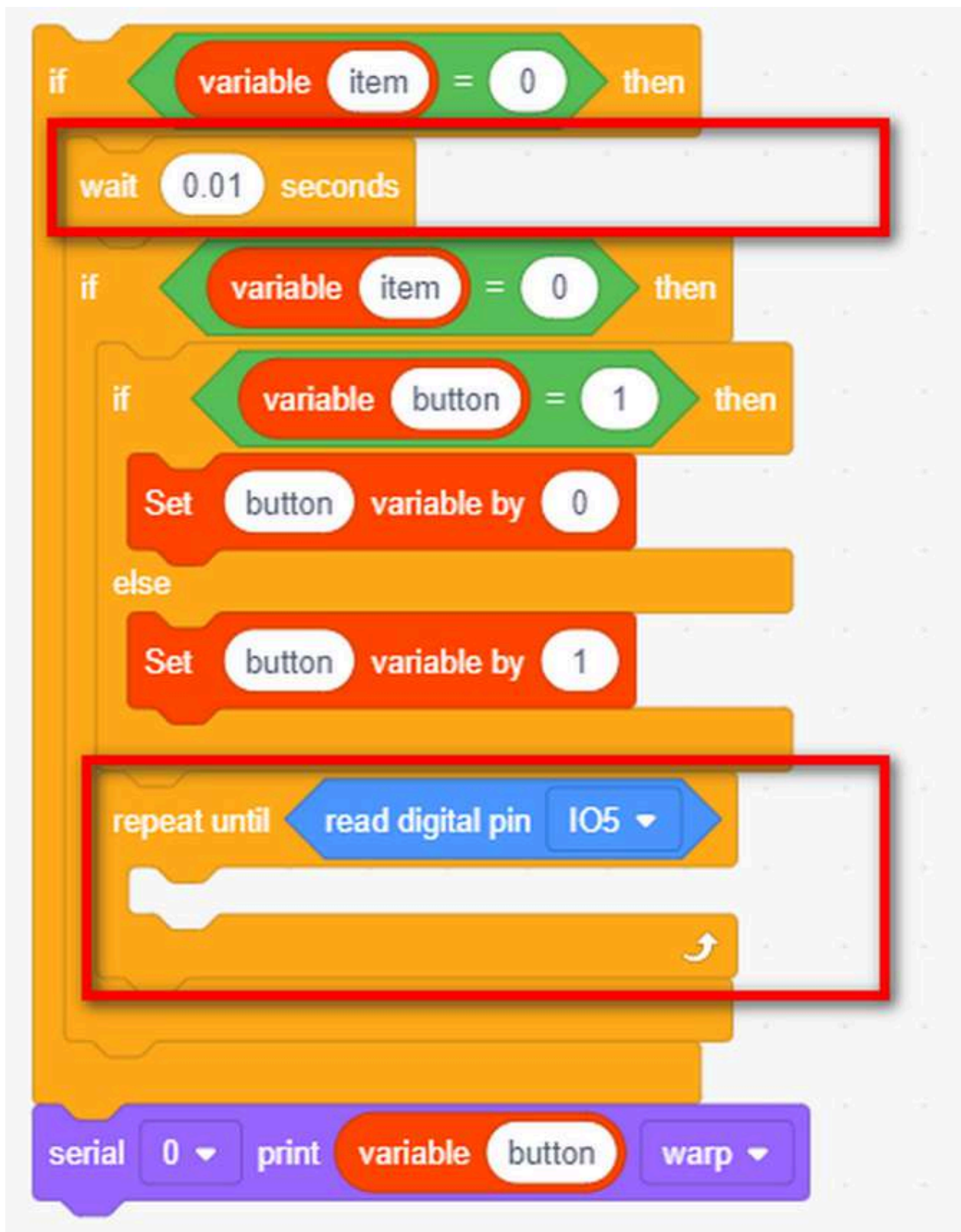
- Definiți două variabile: **item** ca valoare a butonului de citire și **butonul** ca valoare deplasată de buton.



- Atribuiți valoarea butonului de citire **elementului**.



- Determinați dacă butonul este apăsat. Dacă este, schimbați valoarea **butonului** și imprimați-l.



- Întârziere de 0,01 secunde pentru a elimina fluctuația butonului.
 - Dacă este detectată o stare de închidere la buton, se va produce o întârziere executat pentru a elimina **Front Porch Jitter**. În general, întârzierea este de 5 ms ~ 10 ms (Proprietățile mecanice decid). După Fluctuația dispăre, verificați din nou starea butonului. Dacă nivel de stat închis este încă menținut, se confirmă că este apăsat un buton.
 - Când este detectat un buton eliberat, o întârziere de 5 ms ~ 10 m: întâmple să îndepărteze **fluctuația verandei din spate**, astfel încât poate fi executat.

- Când butonul este apăsat, **butonul** este egal cu 1. Apăsați-l din nou, **butonul** se schimbă la 0, alternativ.

Cod complet:

when Arduino begin

serial 0 begin baudrate 115200

set pin IO5 mode input

Declare Global variable Type int Name item Assigned to 0

Declare Global variable Type int Name button Assigned to 0

forever

if read digital pin IO5 then

Set item variable by 1

else

Set item variable by 0

if variable item = 0 then

wait 0.01 seconds

if variable item = 0 then

if variable button = 1 then

Set button variable by 0

else

Set button variable by 1

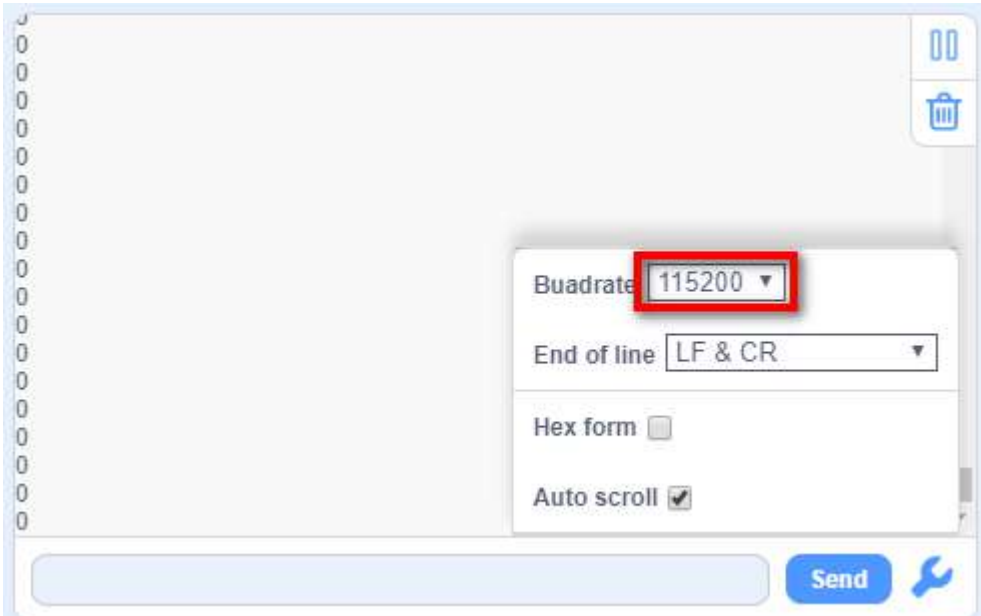
repeat until read digital pin IO5

serial 0 print variable button warp

Rezultatul testului

Încărcați codul și deschideți monitorul serial.

Când apăsați butonul o dată, va fi afișat 1. Dacă apăsați butonul Pentru a doua oară, valoarea devine 0. Acum, un buton comun se laudă funcția unuia cu autoblocare.



4.1.3 Controlul iluminatului

Descriere

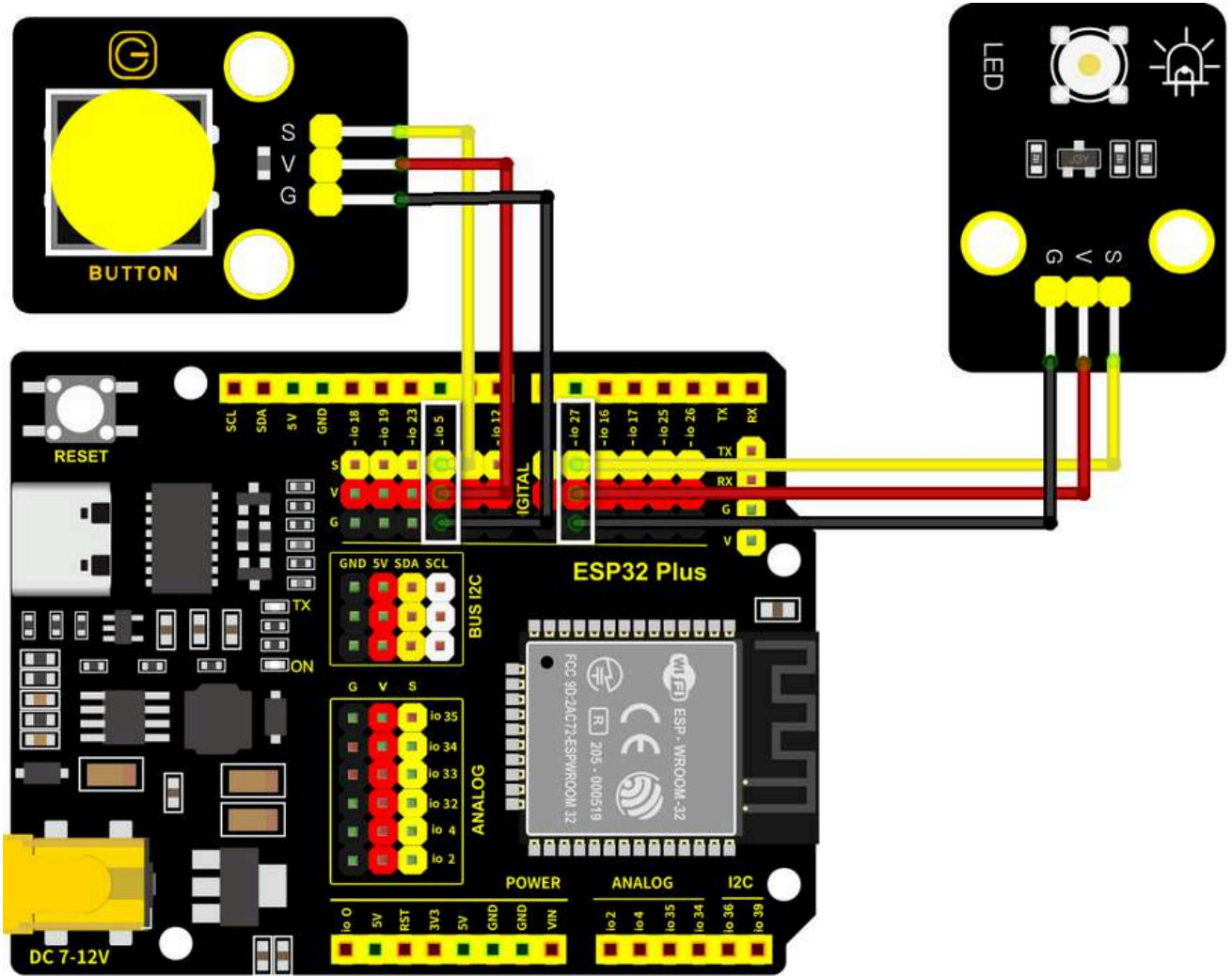
În experimentele de bază de mai sus, remodelăm un buton de blocare automată pentru a controla LED-ul. Un buton de blocare automată este potrivit pentru orice situații în care un o anumită stare trebuie menținută, de exemplu, atunci când LED-ul trebuie pentru o lungă perioadă de timp, placa de dezvoltare ESP32 este necesară pentru unele operațiuni.

În acest experiment, vom adopta placa Arduino ESP32 pentru a vă ghida către implementați un sistem de iluminare și simulați scene din viața reală pentru a controla lumină prin buton.

Schema:

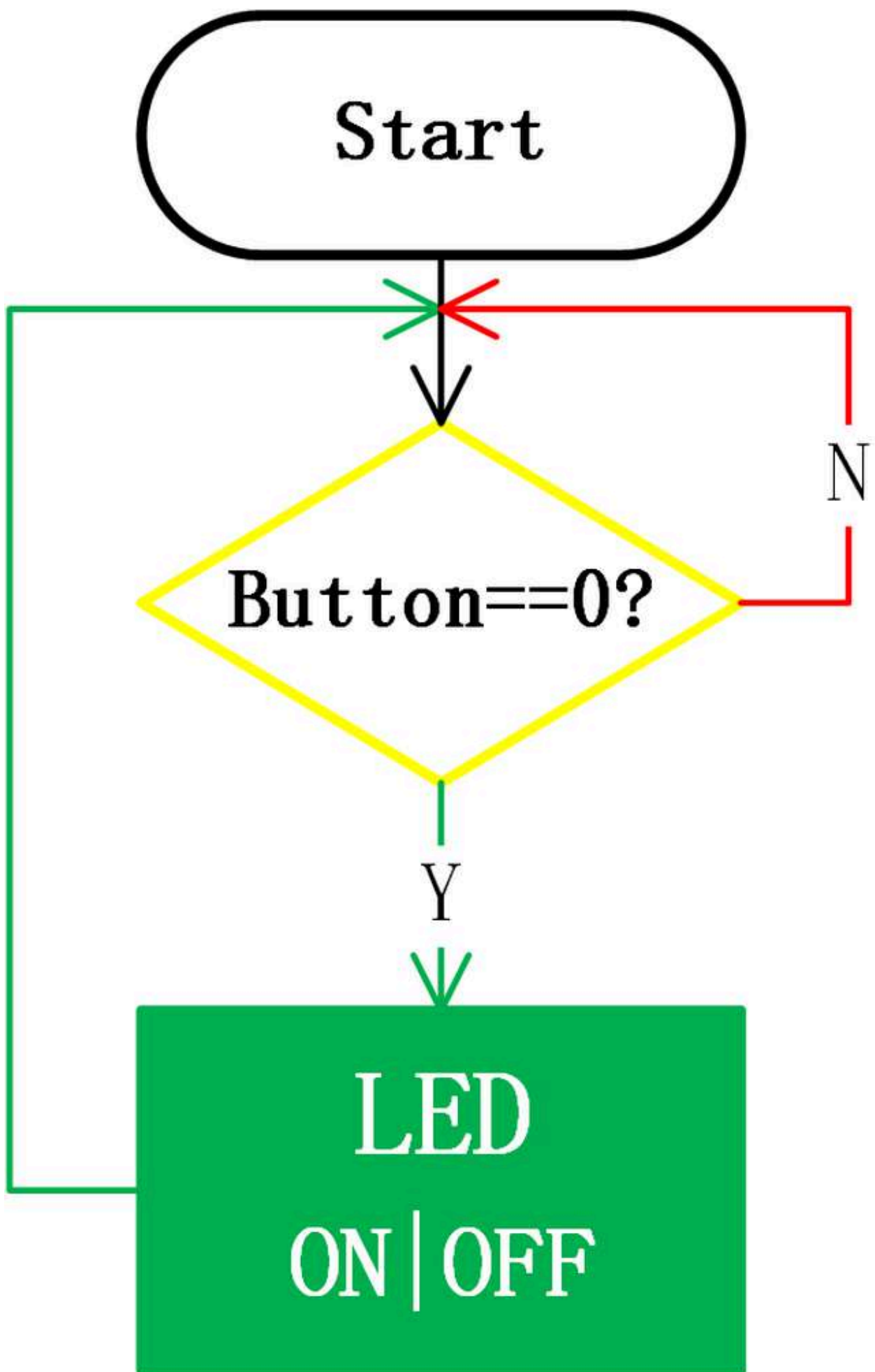
Conectați butonul la io5 și LED-ul la io27

Atenție: Conectați galben la S (Semnal), roșu la V (Alimentare) și negru la GND inversați!



Cod de testare:

Flux de cod:



Cod complet:

 [Ultima](#) ▼

Pe baza codului pentru butonul de blocare automată, adăugăm "ieșire pini LED" Blocuri.

when Arduino begin

serial 0 begin baudrate 115200

set pin IO5 mode input

Declare Global variable Type int Name item Assigned to 0

Declare Global variable Type int Name button Assigned to 0

forever

if read digital pin IO5 then

Set item variable by 1

else

Set item variable by 0

if variable item = 0 then

wait 0.01 seconds

if variable item = 0 then

if variable button = 1 then

Set button variable by 0

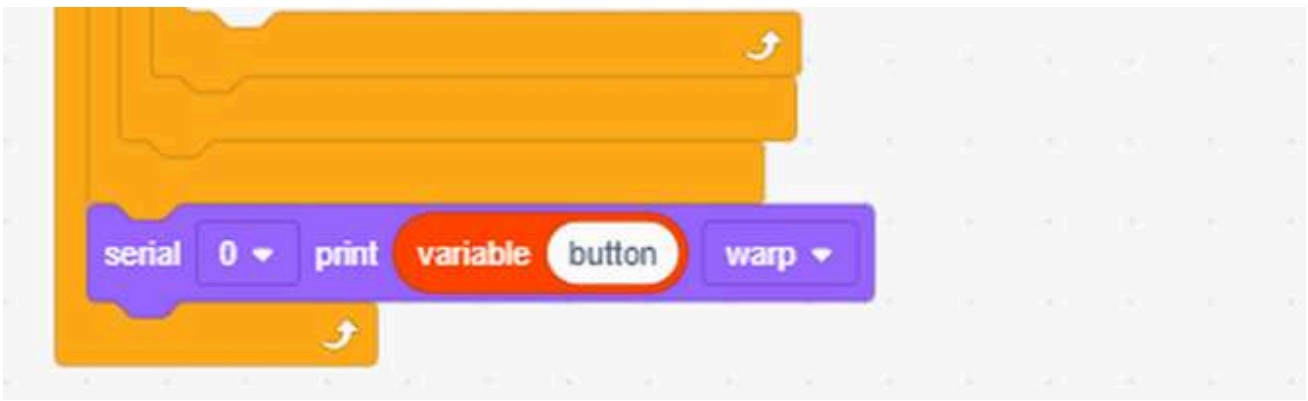
LED pin IO27 output LOW

else

Set button variable by 1

LED pin IO27 output HIGH

repeat until read digital pin IO5



Rezultatul testului:

Când apăsați butonul o dată, LED-ul se aprinde; dacă apăsați din nou, LED-ul Dezactivează. Această operațiune este o buclă, care este în concordanță cu Principiul de iluminare în realitate.

În acest capitol, am demonstrat cum să programăm și să controlăm prin KidsBlock și am învățat elementele de bază, precum și unele programe și concepte hardware în experimente, cum ar fi butonul de blocare automată și sistem de control al iluminatului.

Acestea sunt esențiale pentru un bun dezvoltator KidsBlock. În continuare, vom ghida pentru a continua să explorați mai multe aplicații și abilități, indiferent dacă sunteți un începător sau veteran. Sper să vă bucurați de distracție și provocări în timpul învățarea KidsBlock. Să mergem mai departe!

4.1.5 ÎNTREBĂRI FRECVENTE

Î: LED-ul nu se aprinde după încărcarea codului.

R: Vă rugăm să verificați dacă pinul definit în cod este în concordanță cu acesta în cablurile tale. Dacă sunt incompatibile, vă rugăm să o ajustați făcând referire la codul.

Î: Butonul uneori funcționează, alteori nu.

R: Vă rugăm să modificați întârzierea eliminării jitter-ului la o valoare adecvată.

```
//Eliminate the button jitter  
delay(10); //Modify the delay val in this line
```

4.2 Proiect: Sistem de control al luminii

În acest proiect, vom construi un sistem de control al luminii printr-un fotorezistor și un LED. Este un sistem inteligent de reglare a luminii, care economisește energie și sporește eficiența.

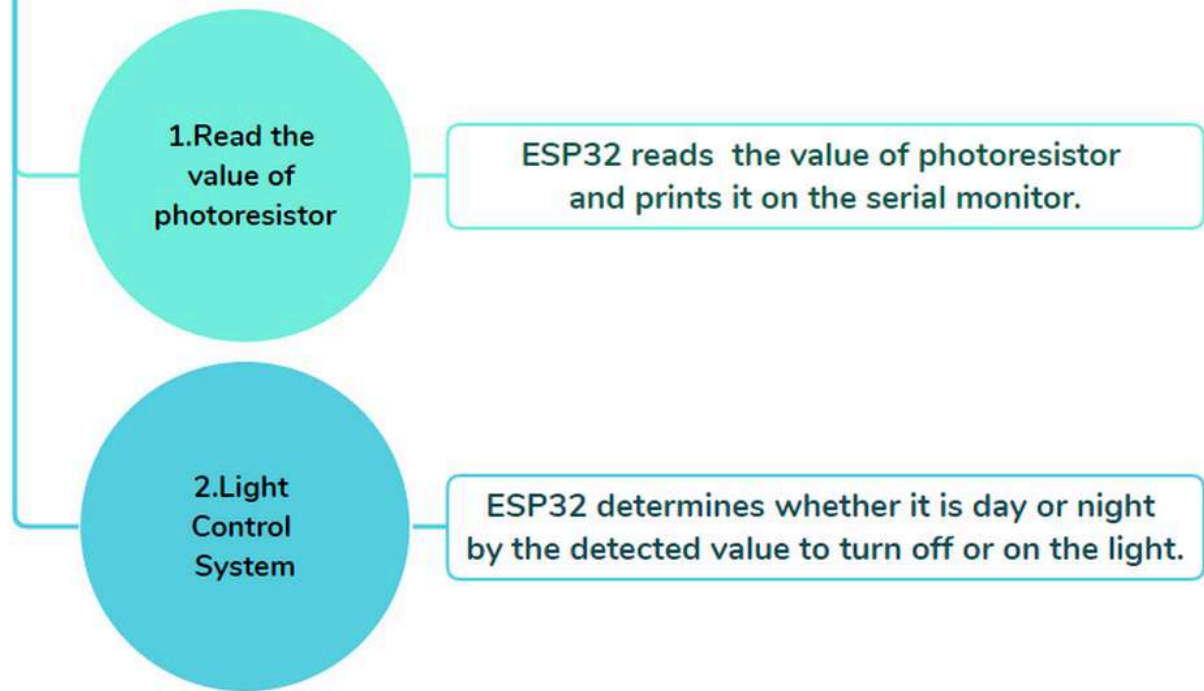


Acest sistem este compatibil cu mai multe condiții. Datorită fotorezistor, este capabil să detecteze intensitatea luminii în timpul zilei sau noaptea, realizând un sistem mai inteligent și mai economisitor de energie.

Când fotorezistorul detectează că luminozitatea ambientală este mai mică decât setați valoarea, LED-ul se aprinde. Dimpotrivă, dacă lumina ambientală este mai mare decât valoarea setată, fotorezistorul va trimite un semnal diferit pentru a opri LED-ul.

4.2.1 Diagrama de flux

Light Control System



4.2.2 Fotorezistență

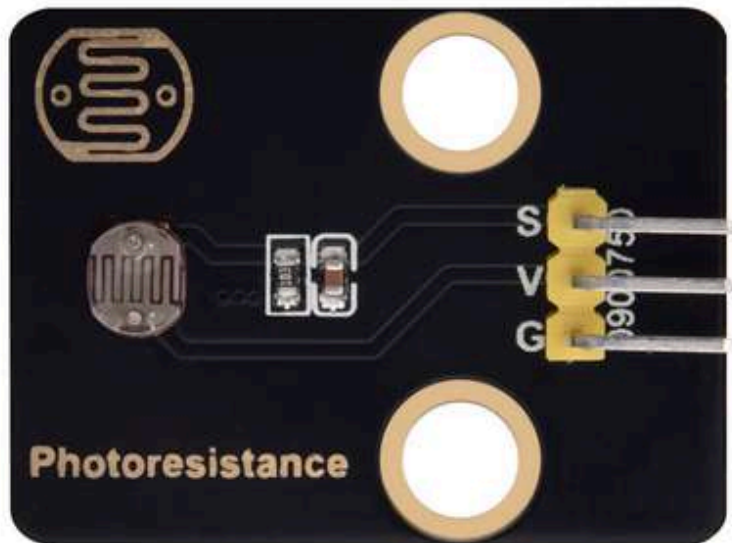
Descriere:

Un fotorezistor, numit și fotosenzor, transformă semnalul luminos în semnal electric (tensiune, curent și rezistență).

Principiul de funcționare:

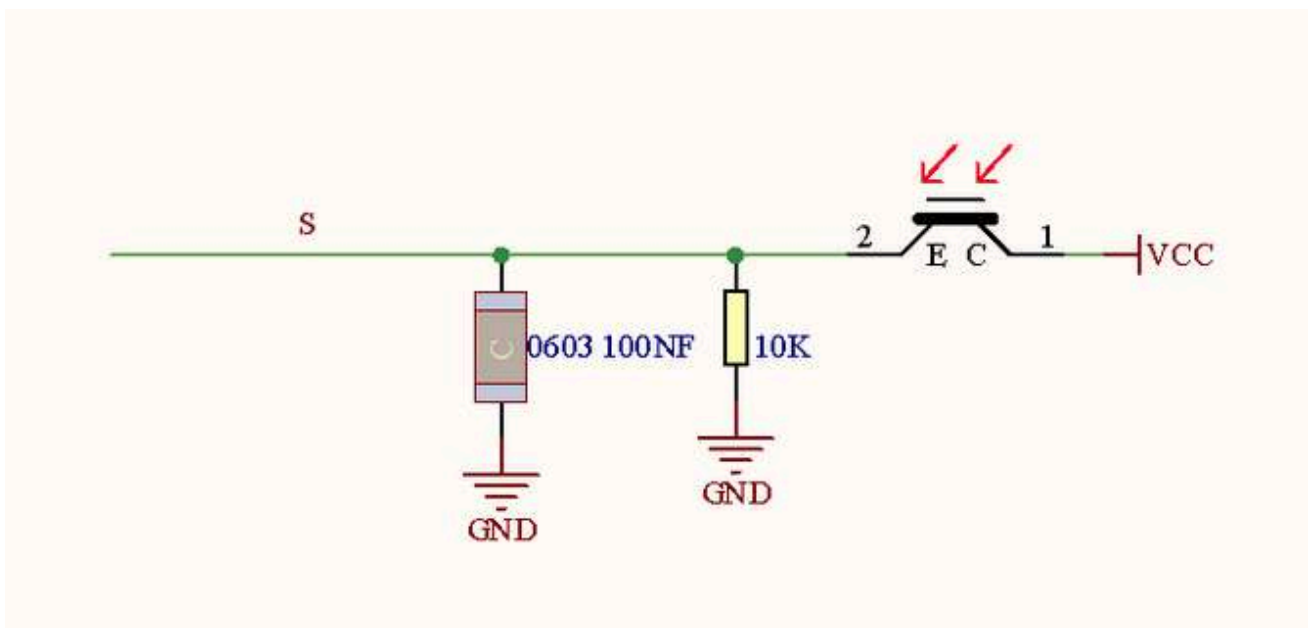
Plasăm un fotorezistor într-un circuit în conexiune în serie și adăugăm tensiune adecvată pentru ambii poli. Când nu există lumină, rezistența este infinit și circuitul aproape că se deschide. Cu toate acestea, când este lumină, rezistența scade în timp ce curentul crește și este echivalent cu un scurtcircuit atunci când intensitatea luminii este suficientă.

Acum vom citi valoarea fotorezistorului prin programare pe ESP32 consiliul de dezvoltare.



Diagramă schematică:

Când lumina lovește fotorezistorul, cu cât lumina este mai puternică, Rezistența va fi mai mică, deci cu cât tensiunea VCC va trece mai mare prin rezistență.



Parametrii:

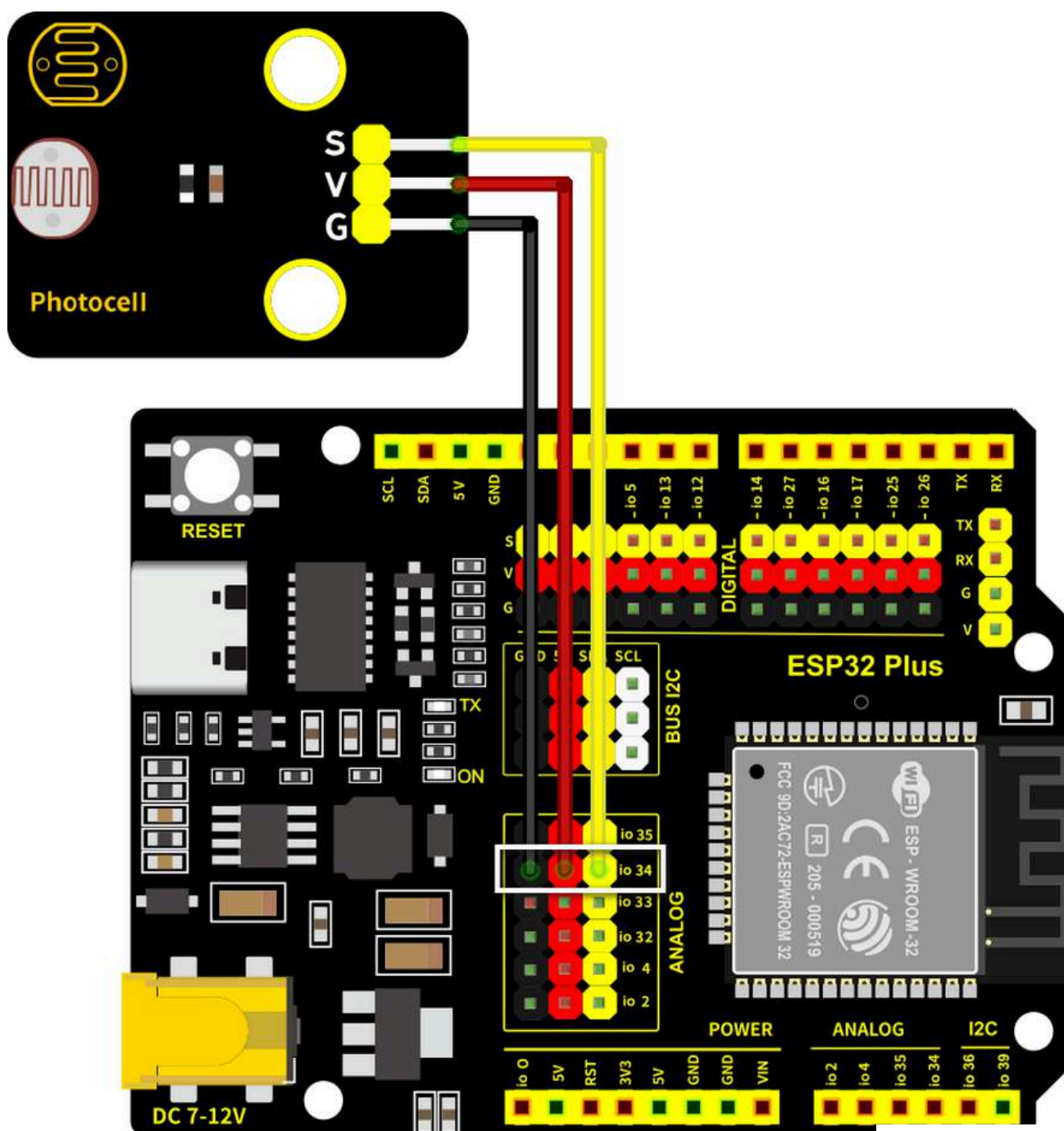
- Tensiune: 3 ~ 5V
- Curent: 0,2 mA

- Putere: 1mW
- Valoarea de vârf a spectrului: 540 nm
- Rezistență la lumină (10lux): 5 ~ 10KR
- Rezistență la întuneric: 0.5MR

Schema:

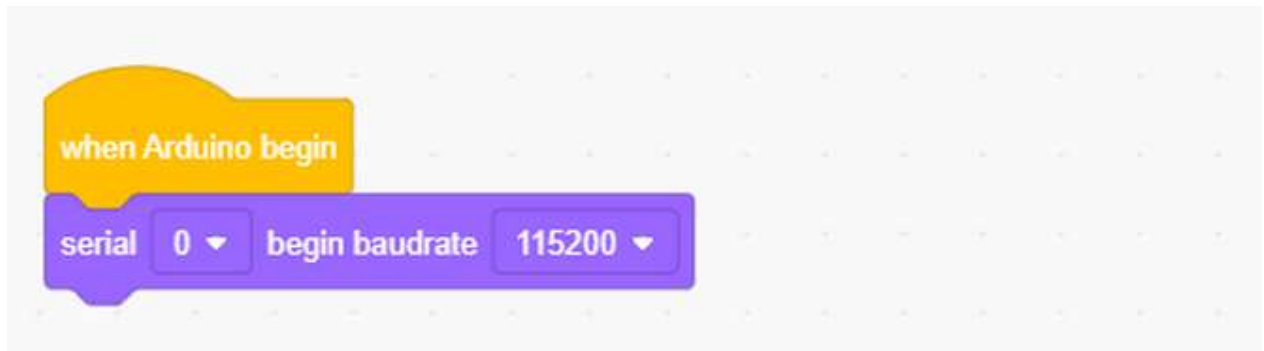
Conectați fotorezistorul la io34.

Atenție: Conectați galben la S (Semnal), roșu la V (Alimentare) și negru la GND. Nu le inversați!

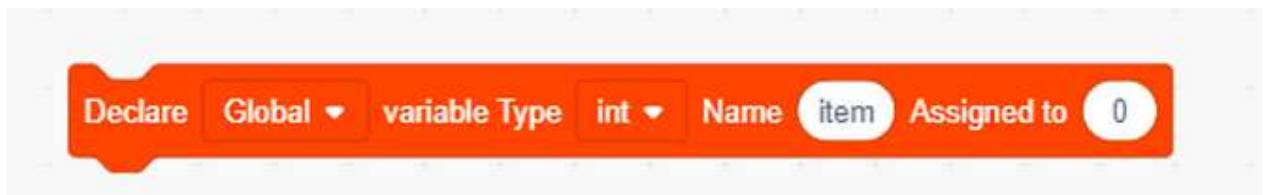


Cod de testare:

- Inițializați portul serial.



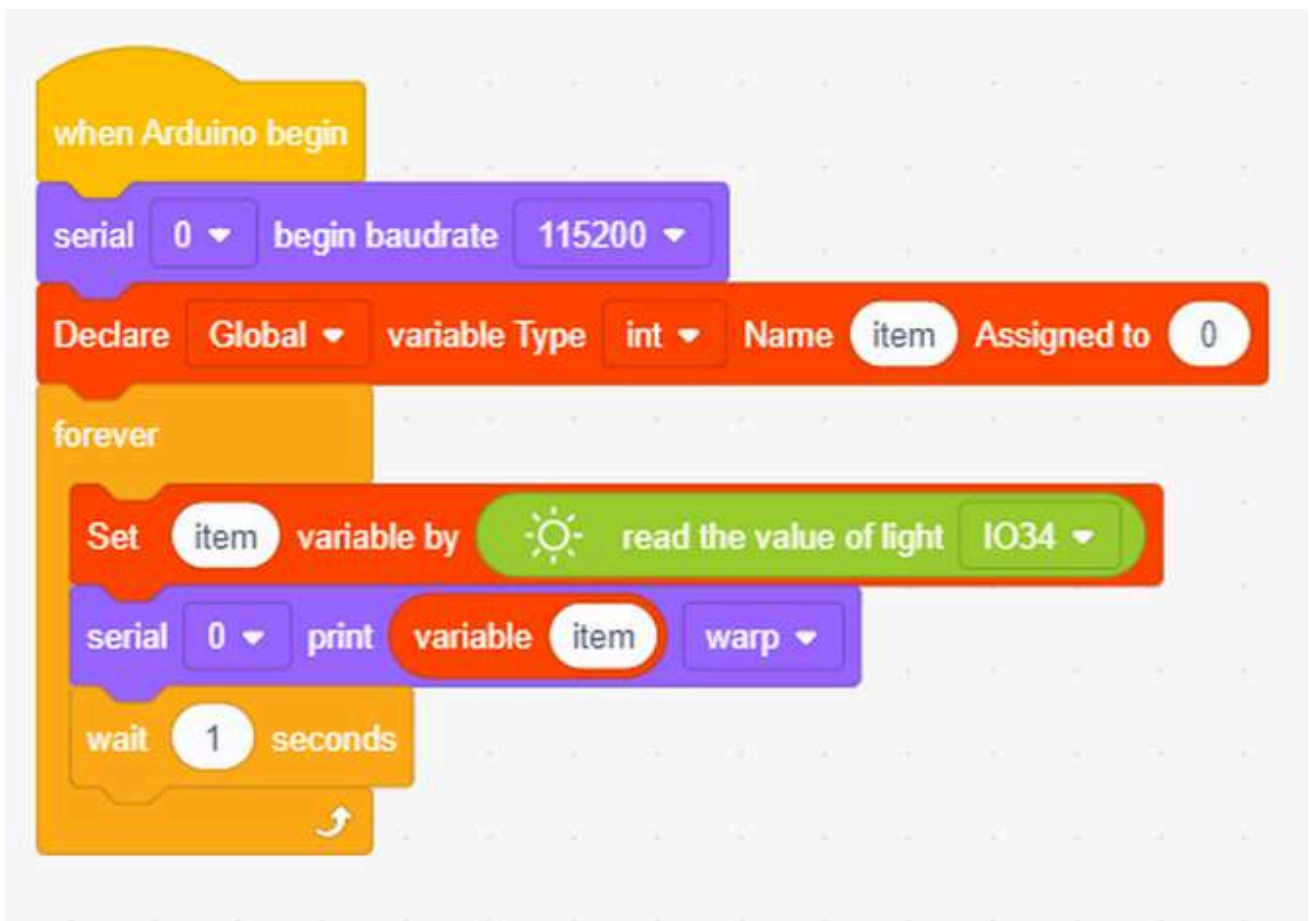
- Definiți o variabilă globală "item" ca valoare a fotorezistorului.



- Setați "item" la valoarea citită și imprimați-l pe monitorul serial.



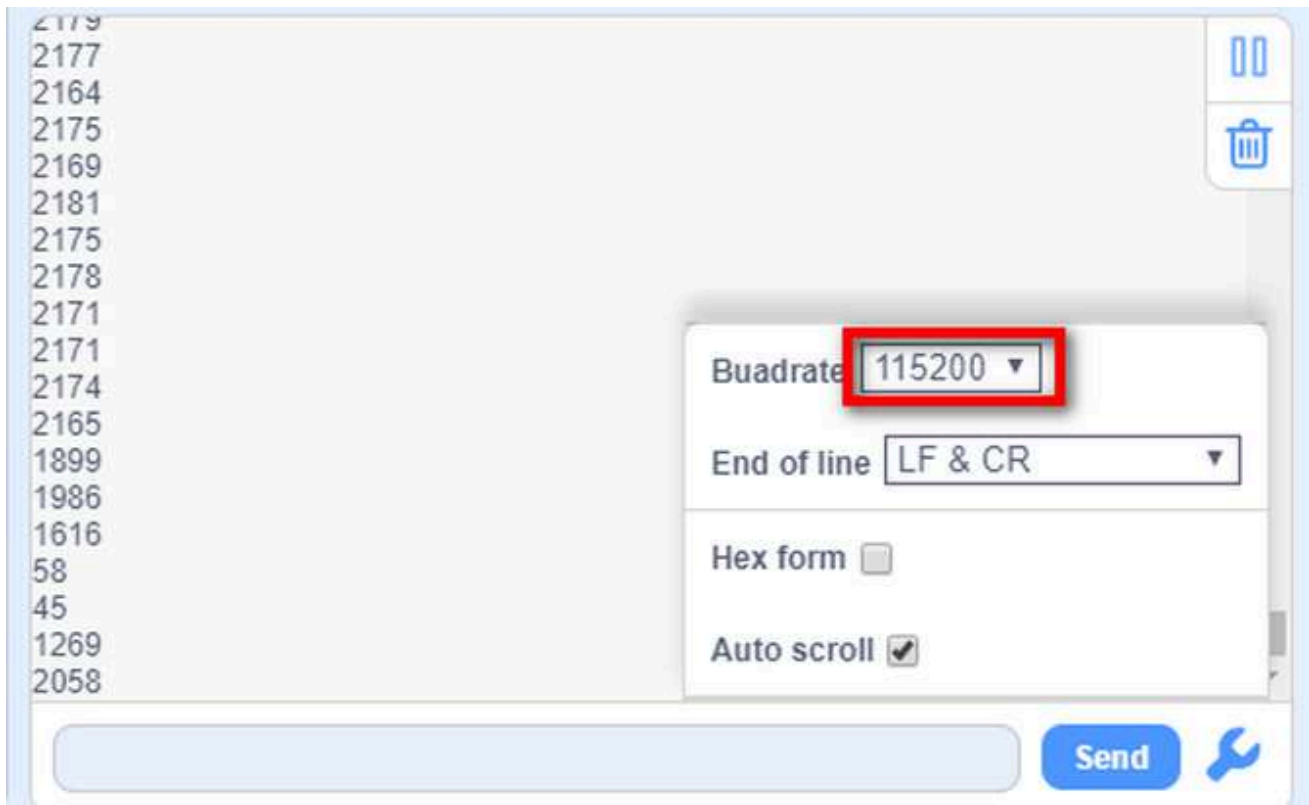
Cod complet:



Rezultatul testului:

Deschideți monitorul serial.

Cu cât lumina detectată de fotorezistor este mai puternică, cu atât este mai mare valoarea va fi.

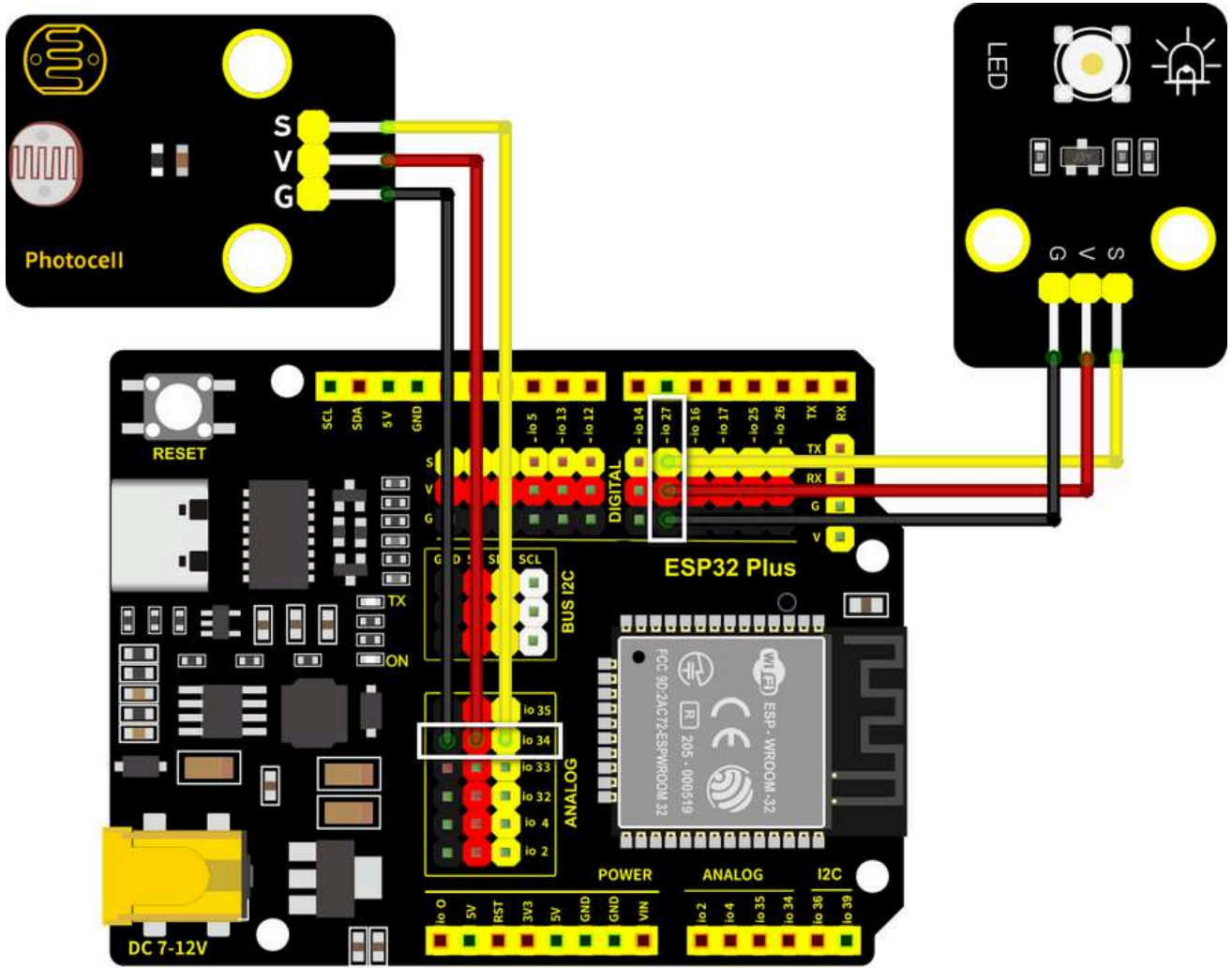


4.2.3 Sistem de control al luminii

Schema:

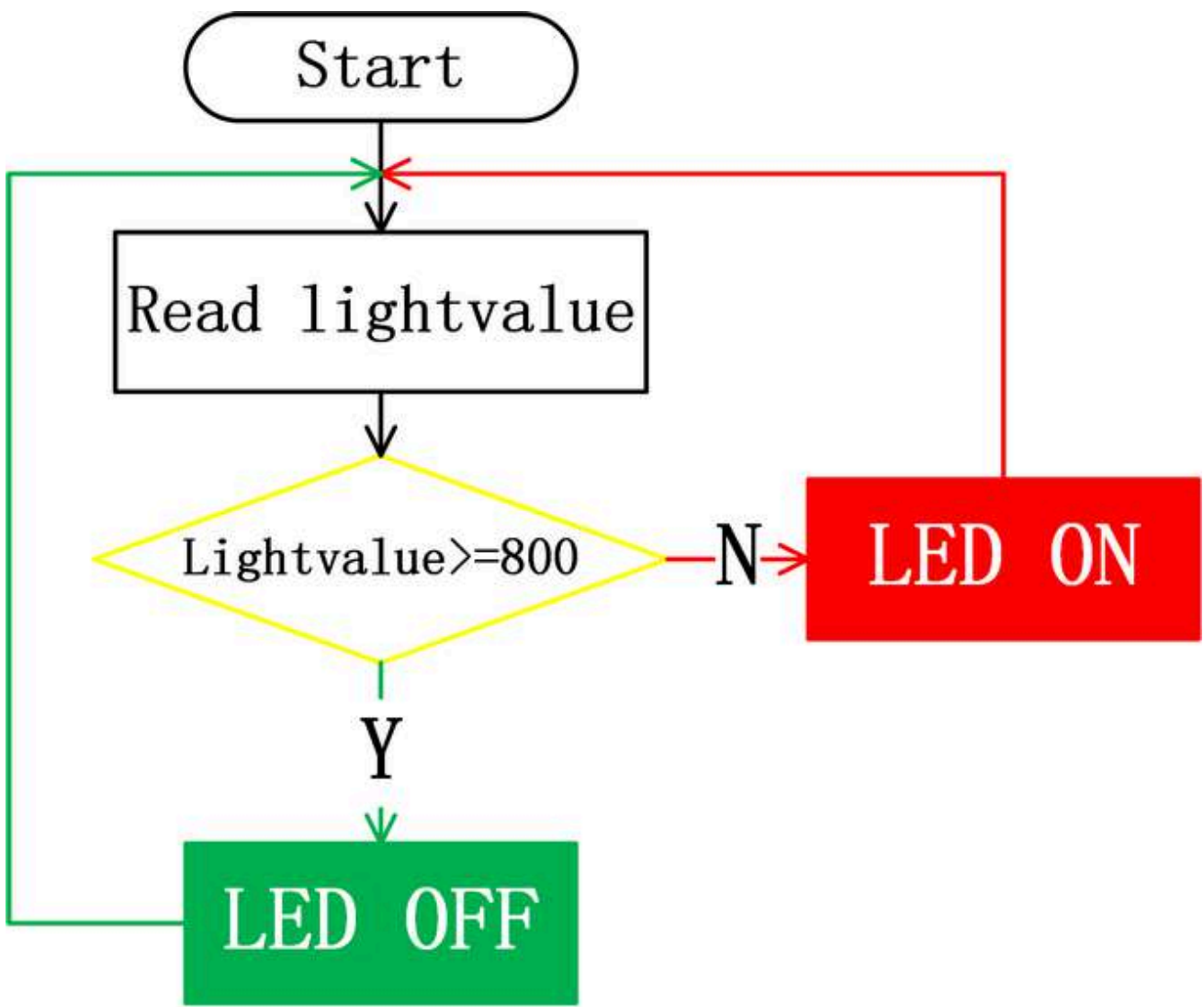
Conectați fotorezistorul la io34 și LED-ul la io27.

Atenție: Conectați galben la S (Semnal), roșu la V (Alimentare) și negru la GND. Nu le inversați!



Cod de testare:

Flux de cod:

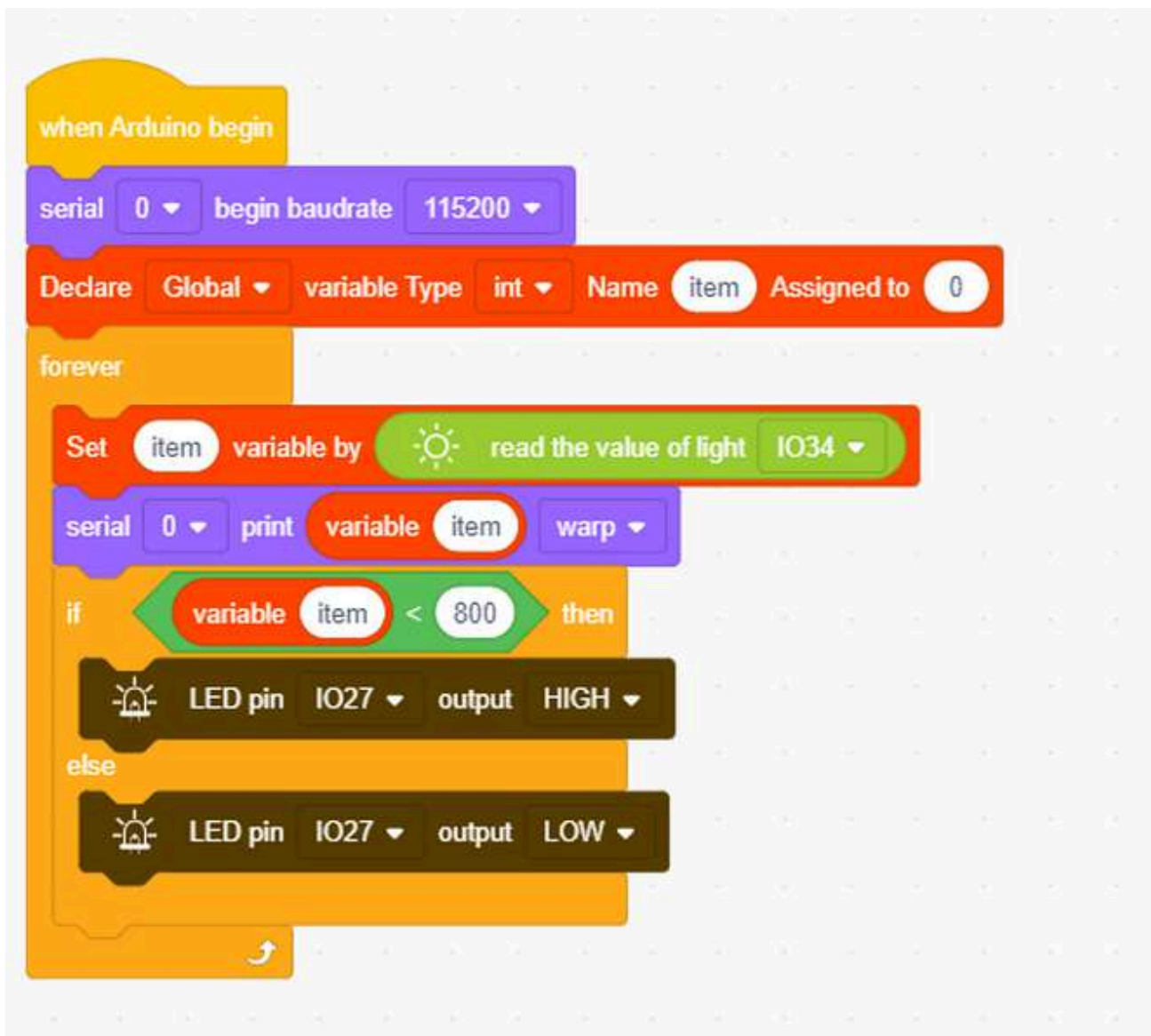


- Determina:
 - Valoarea fotorezistorului ≥ 800 , LED-ul se stinge.
 - Valoarea fotorezistorului ≤ 800 , LED-ul se aprinde.

```

if (variable item < 800) then
  LED pin IO27 output HIGH
else
  LED pin IO27 output LOW
  
```

Cod complet:



Rezultatul testului:

Când valoarea fotorezistorului este mai mare de 800 (în timpul zilei), LED-ul se stinge. Cu toate acestea, dacă valoarea este mai mică de 800, LED-ul va se aprinde automat.



**Diverse condiții pot adopta acest tip de sistem. Datorită fotorezistor, este capabil să detecteze intensitatea luminii în timpul zilei sau noaptea, care economisește energie și intelectualizează întregul sistem. **

4.2.2 ÎNTREBĂRI FRECVENTE

Î: Valoarea fotorezistorului nu poate fi 0.

R: În viața reală, există puțină lumină, deși stingi toate luminile în camera dvs., astfel încât valoarea fotorezistorului se apropie doar de 0 mai degrabă decât este egal cu 0.

Î: După încărcarea codului, LED-ul nu se aprinde chiar dacă camera este întuneric fără lumini.

R: Creșteți valoarea determinată a fotorezistorului. În exemplul nostru, setați la 800. Deci, îl puteți ajusta la 1000 sau o valoare mai mare.

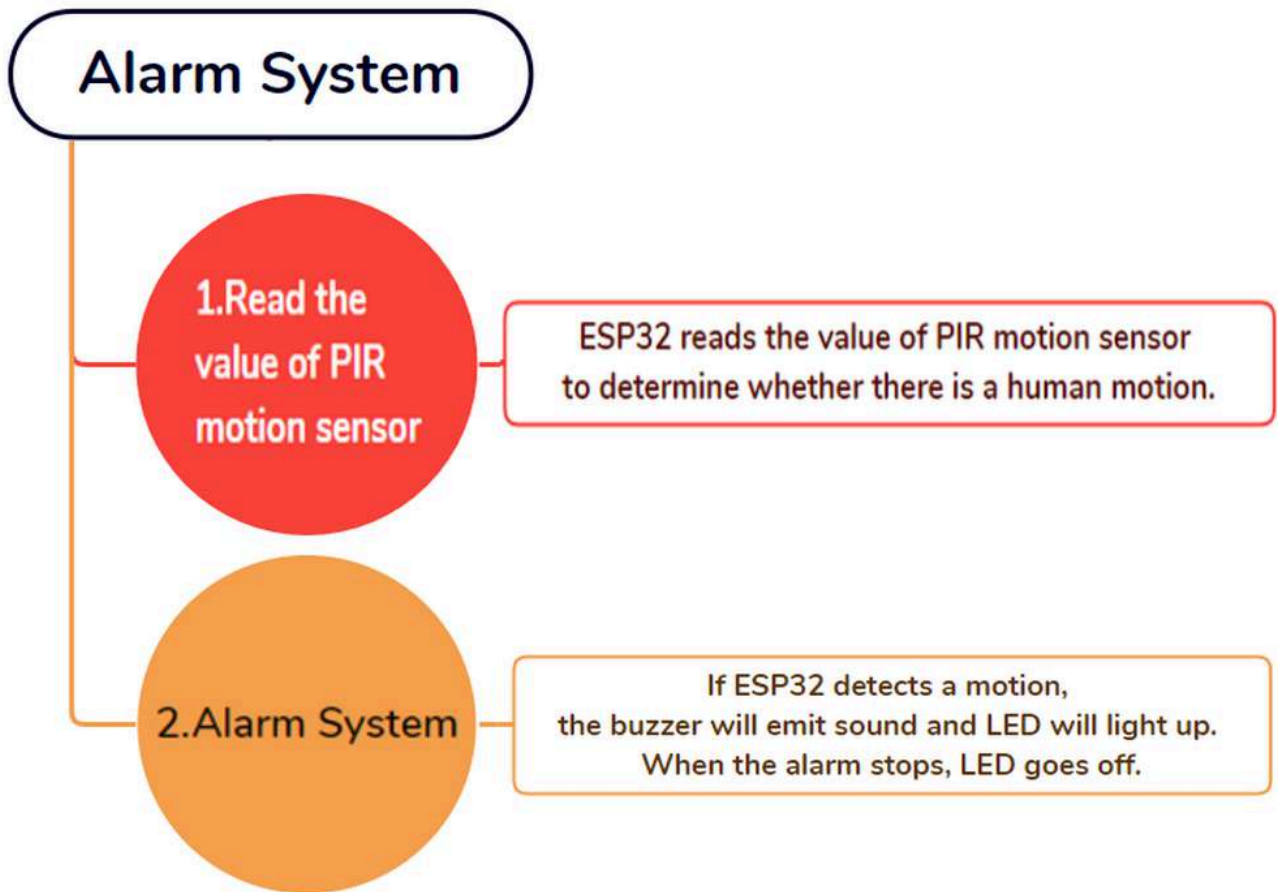


4.3 Proiect: Sistem de alarmă

În acest proiect, folosim un senzor de mișcare PIR și un sonerie pentru a consta într-un sistem de alarmă, care poate fi controlat de placa de dezvoltare ESP32.

Cum funcționează? Semnalele electrice sunt detectate și citite de PIR senzor de mișcare prin programare pe Arduino IDE, apoi determină dacă există o persoană. Dacă există, soneria alarmează. În acest fel, Acest sistem de alarmă costă mult mai puțin pentru familii și birouri.

4.3.1 Diagrama de flux

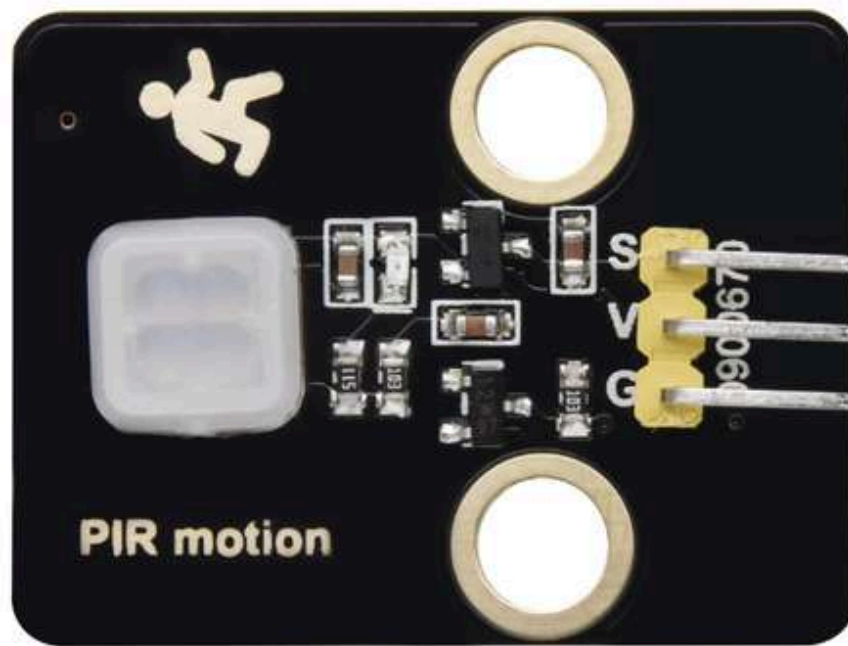


4.3.2 Senzor de mișcare PIR

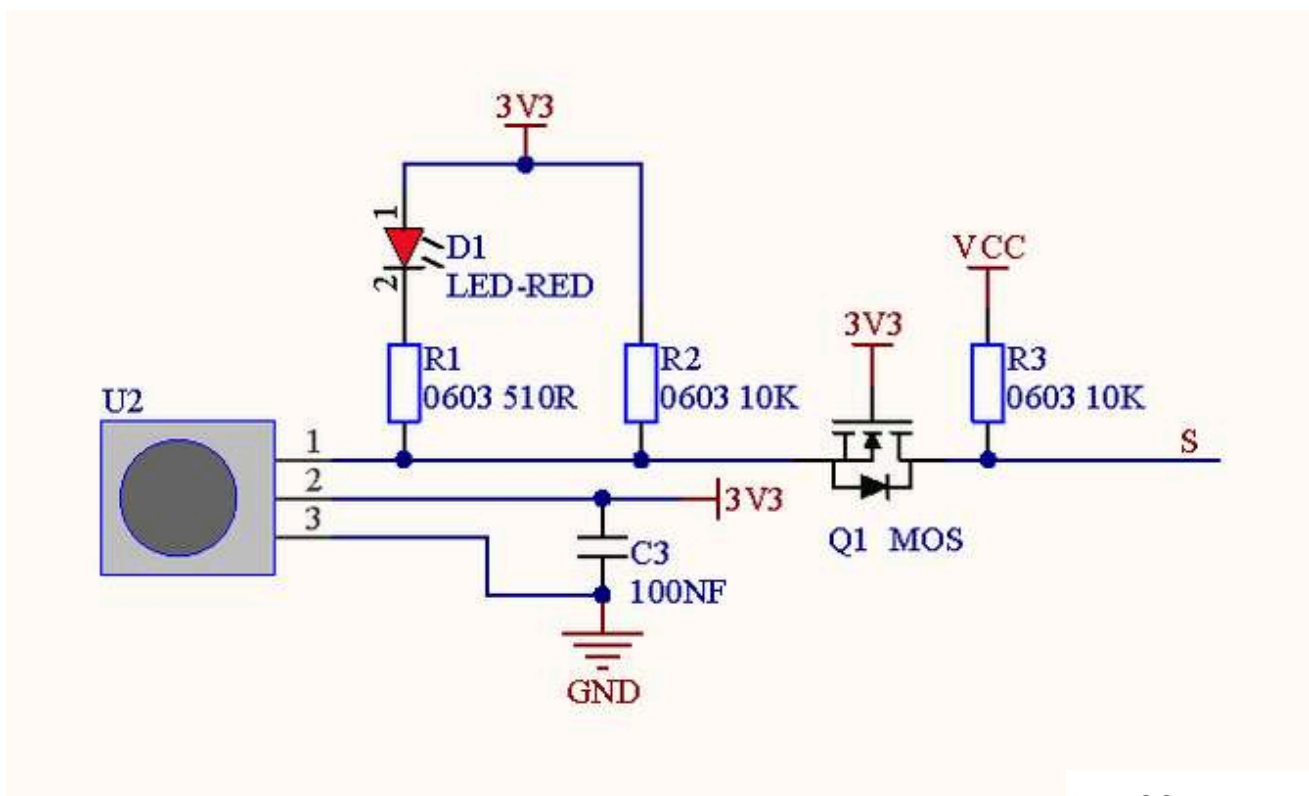
Descriere:

Un senzor de mișcare PIR detectează prezența unei persoane prin detectarea căldurii eliberat de corpul uman.

Mai mult, acest senzor este mic și ușor de utilizat.



Diagramă schematică:



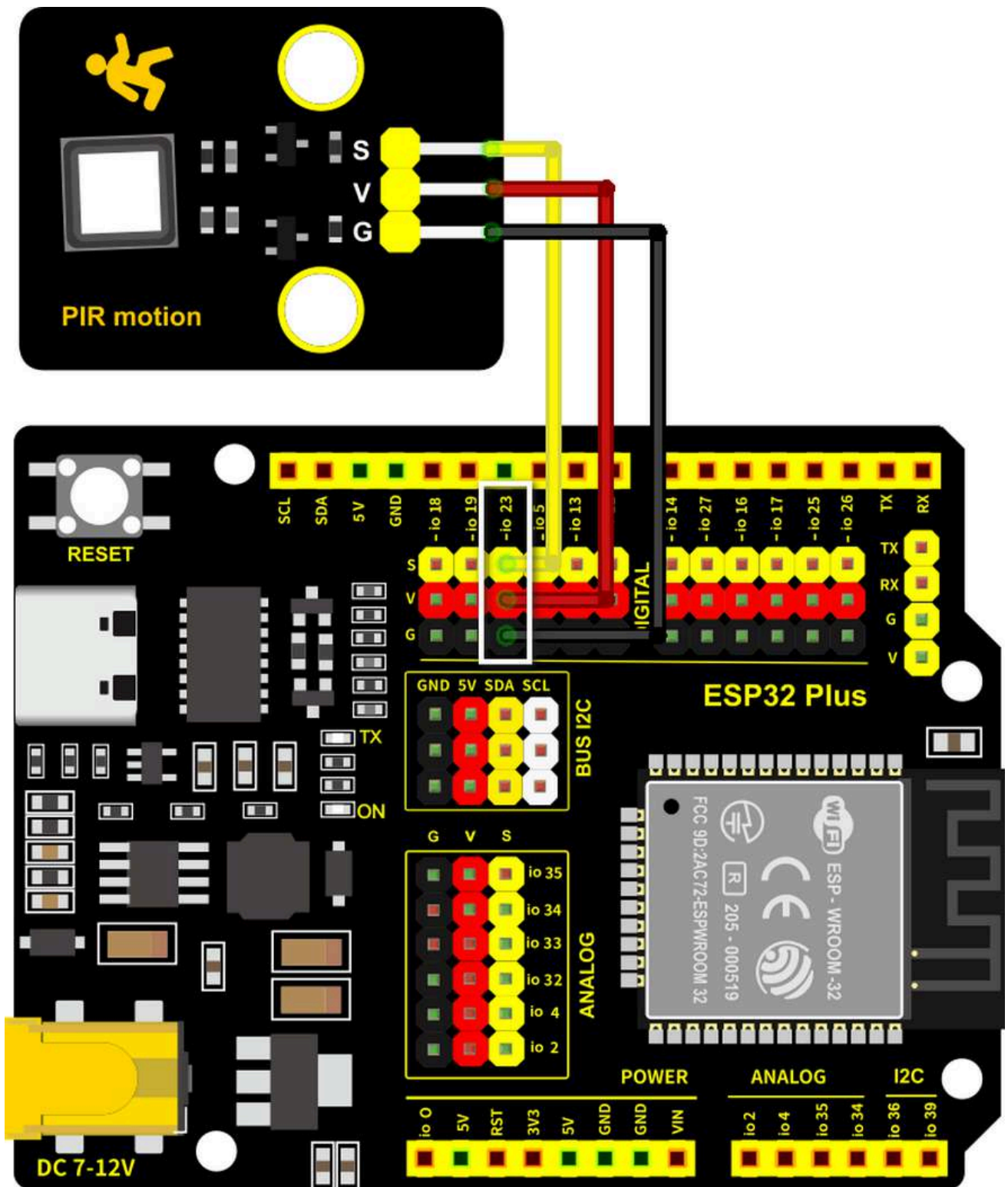
Parametrii:

- Tensiune: 3 ~ 5V
 - Curent: 3.6mA
 - Putere: 18mW
 - Unghi de vizualizare: $Y = 90^\circ$, $X = 110^\circ$ (valoare teoretică)
 - Distanța de detectare: $\leq 5\text{m}$
-

Schema:

Conectați senzorul de mișcare PIR la io23.

Atenție: Conectați galben la S (Semnal), roșu la V (Alimentare) și negru la GND. Nu le inversați!



Cod de testare:

Citiți valoarea de la pinul IO23 pentru a determina dacă există o mișcare umană.

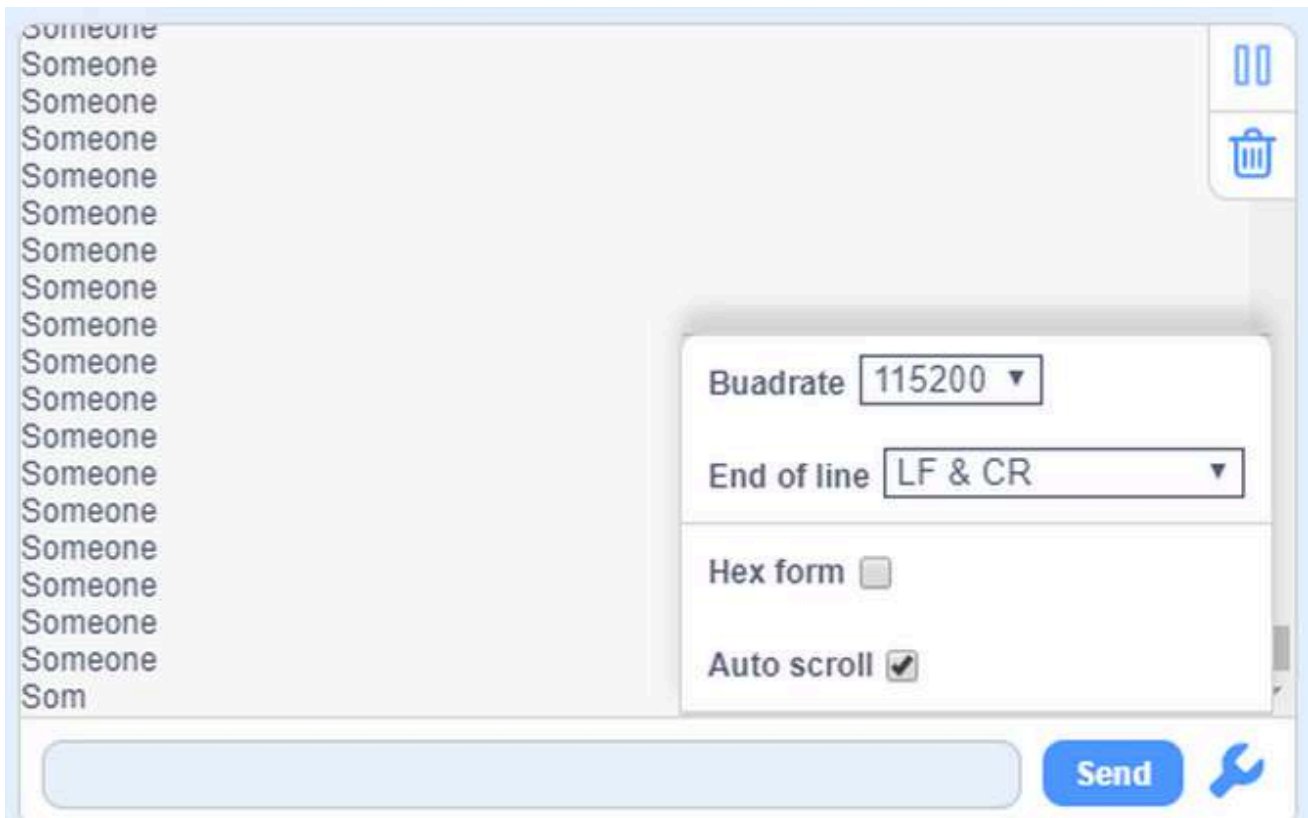


Rezultatul testului:

Deschideți monotorul serial.

Când cineva se află în zonă, **Cineva** este afișat pe monitor, iar LED-ul roșu de pe senzor se stinge. Cu toate acestea, dacă nu este nimeni, **nimeni nu** va fi imprimat și LED-ul va fi întotdeauna aprins.

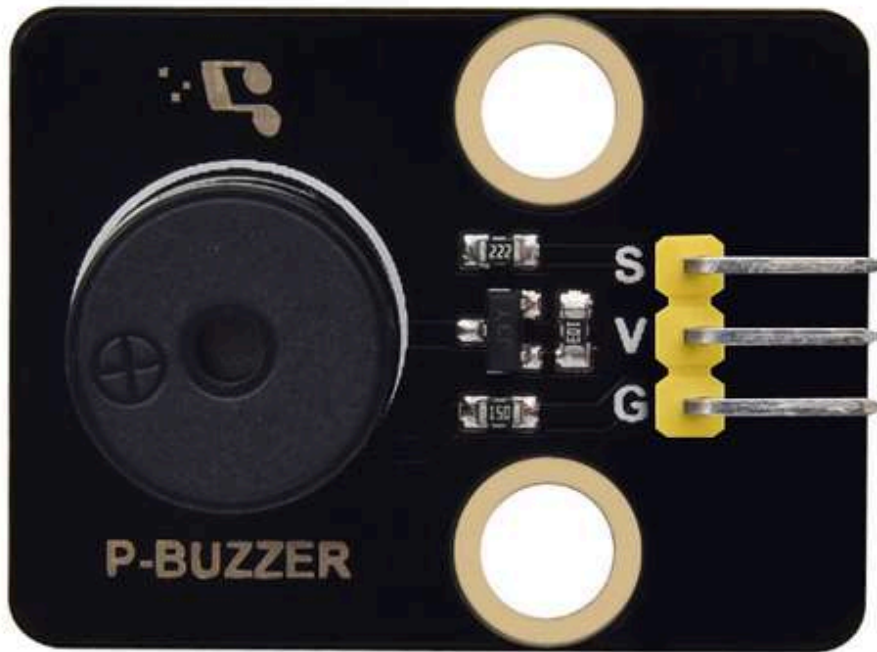
ATENȚIE: Senzorul de mișcare PIR nu este capabil să pătrundă în lucruri, așa că Vă rugăm să nu acoperiți senzorul în timp ce detectați mișcări.



4.3.3 Sonerie

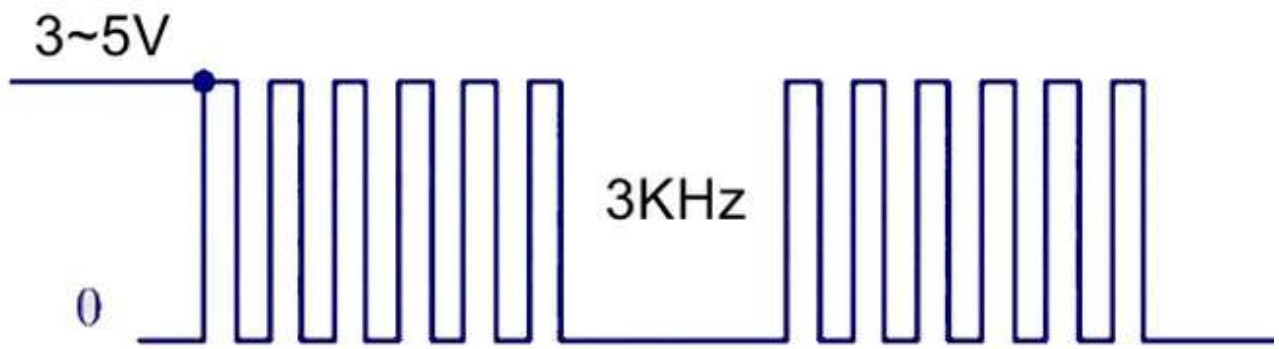
Descriere:

Un buzzer este o sondă electronică, care emite sunete cu diferite frecvențe și durate și este alimentat de tensiune continuă. Astfel, se poate utiliza ca memento sau alarmă în dispozitive electronice considerabile, cum ar fi computere, imprimante, copiatoare, alarme, jucării electronice, electronice, telefoane și temporizatoare.



Un buzzer constă **dintr-un dispozitiv de vibrație** și **un dispozitiv de rezonanță**. Și există două categorii: sonerii pasive și sonerii active.

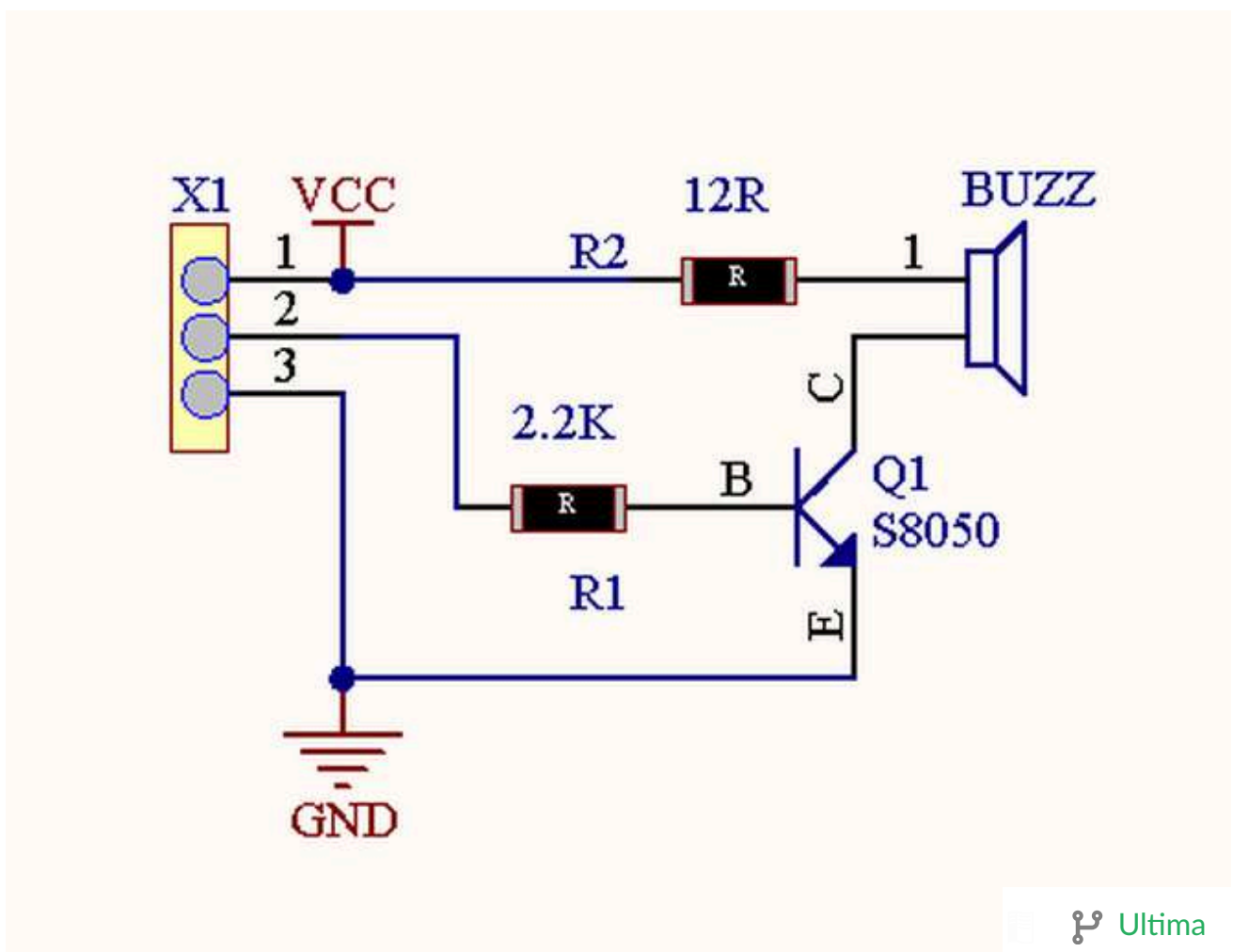
- Un **buzzer pasiv** nu poate emite sunet în sine, decât dacă punând un semnal cu o anumită frecvență. Mai mult, sunetul emis variază din cauza frecvenței diferite a pătratului wave, astfel încât un buzzer pasiv poate simula melodii. `vibrate` `square wave`
 - O undă analogică poate fi generată prin modificarea nivelului de putere la pini. De exemplu, după nivelul ridicat care durează 500ms, trece la un nivel scăzut pentru încă 500 ms, apoi la un nivel ridicat iarăși...
 - ****Conducem soneria printr-o undă de scutier în limita a 200 ~ 5000Hz și poate calcula frecvența (f): $f = 1 / T$; T este perioada (totalul timp de nivel ridicat și scăzut). ****



- Un **buzzer activ** este capabil să emită sunet automat fără un motivator extern, deoarece include un circuit de conducere care numai nevoii. Cu toate acestea, sunetul său este plat cu relativ frecvență fixă. [DC power supply](#)

În acest experiment, un buzzer pasiv este aplicat pentru a "reda muzică".

Diagramă schematică:



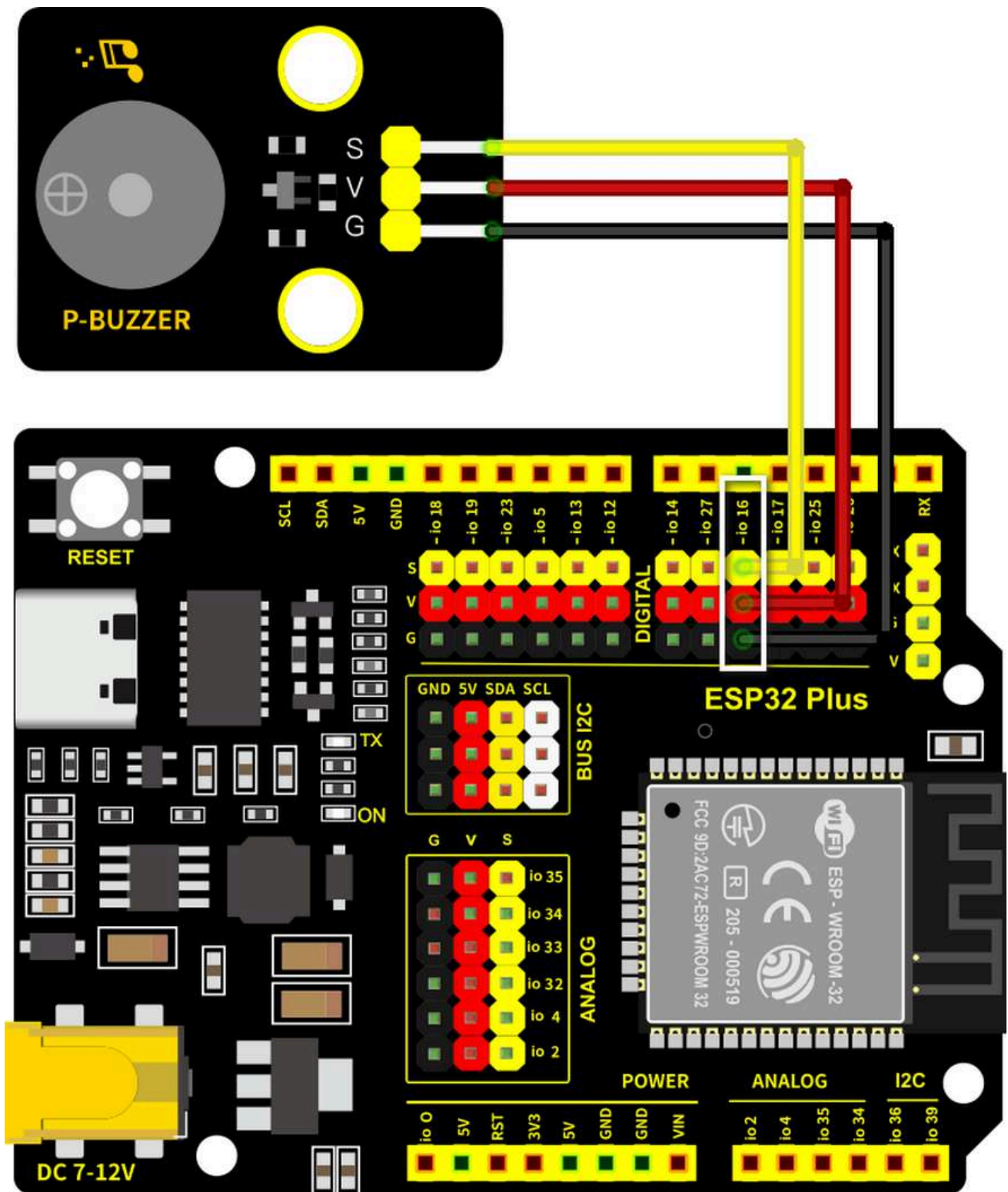
Parametrii:

- Tensiune: 3 ~ 5V
 - Curent: $\leq 5\text{mA}$
 - Putere: $\leq 25\text{mW}$
-

Schema:

Conectați soneria la io16.


Atenție: Conectați galben la S (Semnal), roșu la V (Alimentare) și negru la GND. Nu le inversați!



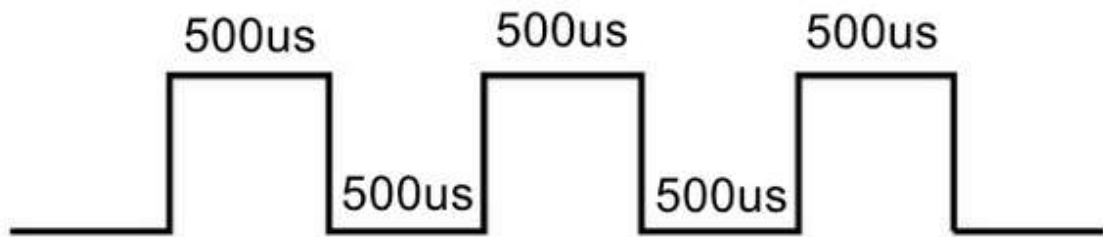
Cod de testare:

Metoda 1: Analog Squire Wave

Un buzzer pasiv este condus de valuri de scutier, așa că stimulăm unda.

O undă analogică poate fi generată prin modificarea nivelului de putere al PI  Ultima ▼ pentru 500us și nivel scăzut pentru 500us. Deci, soneria va emit sunet. De asemenea, duratele pot regla volumul sunetului.

Vă rugăm să încercați 1000us, 1500us, 3000us... Care este diferența?



Cod:

```
when Arduino begin
  set pin IO16 mode output
  forever
    set digital pin IO16 out high
    wait 0.005 seconds
    set digital pin IO16 out low
    wait 0.005 seconds
```

The code block is written in a block-based programming style. It starts with a yellow 'when Arduino begin' block. This is followed by a blue 'set pin IO16 mode output' block. Below that is an orange 'forever' loop block. Inside the loop, there are four blocks: a blue 'set digital pin IO16 out high' block, an orange 'wait 0.005 seconds' block, a blue 'set digital pin IO16 out low' block, and another orange 'wait 0.005 seconds' block. The loop ends with a white arrow icon.

- În funcția de întârziere, unitatea de timp este micro-secunde. Deci, următoarele reprezintă o întârziere de 500 ms.



Conform formulei:

$$f = 1/T$$

Astfel, 500us este durata și putem calcula frecvența = 2kHz, adică nivelul înalt și scăzut se modifică de 2000 de ori pe secundă.

Metoda 2: Blocuri de difuzoare

Adoptăm blocuri de cod pentru difuzoare  pentru a conduce soneria să vibreze.

Blocurile de difuzoare generează semnal PWM cu o anumită frecvență de acționat soneria să vibreze, iar durata și tonul sunt controlate de parametrii aferenți.

Există două moduri de a defini durata. Unul este de a regla parametrii funcției `tone()` pentru a seta o durată, iar celălalt este să adoptați o funcție `noTone()` pentru a opri direct sunetul. Dacă nu definiți o durată în `tone()`, semnalul sonor va fi întotdeauna generat cu excepția cazului în care un `noTone()` îl oprește.

Pentru placa ESP32, un singur sunet poate fi produs la un moment dat. Dacă un pin de ESP32 generează un semnal sonor prin `tone()`, nu este acceptabil să emite sunet prin această funcție pe un alt pin.

Tabel de tonuri

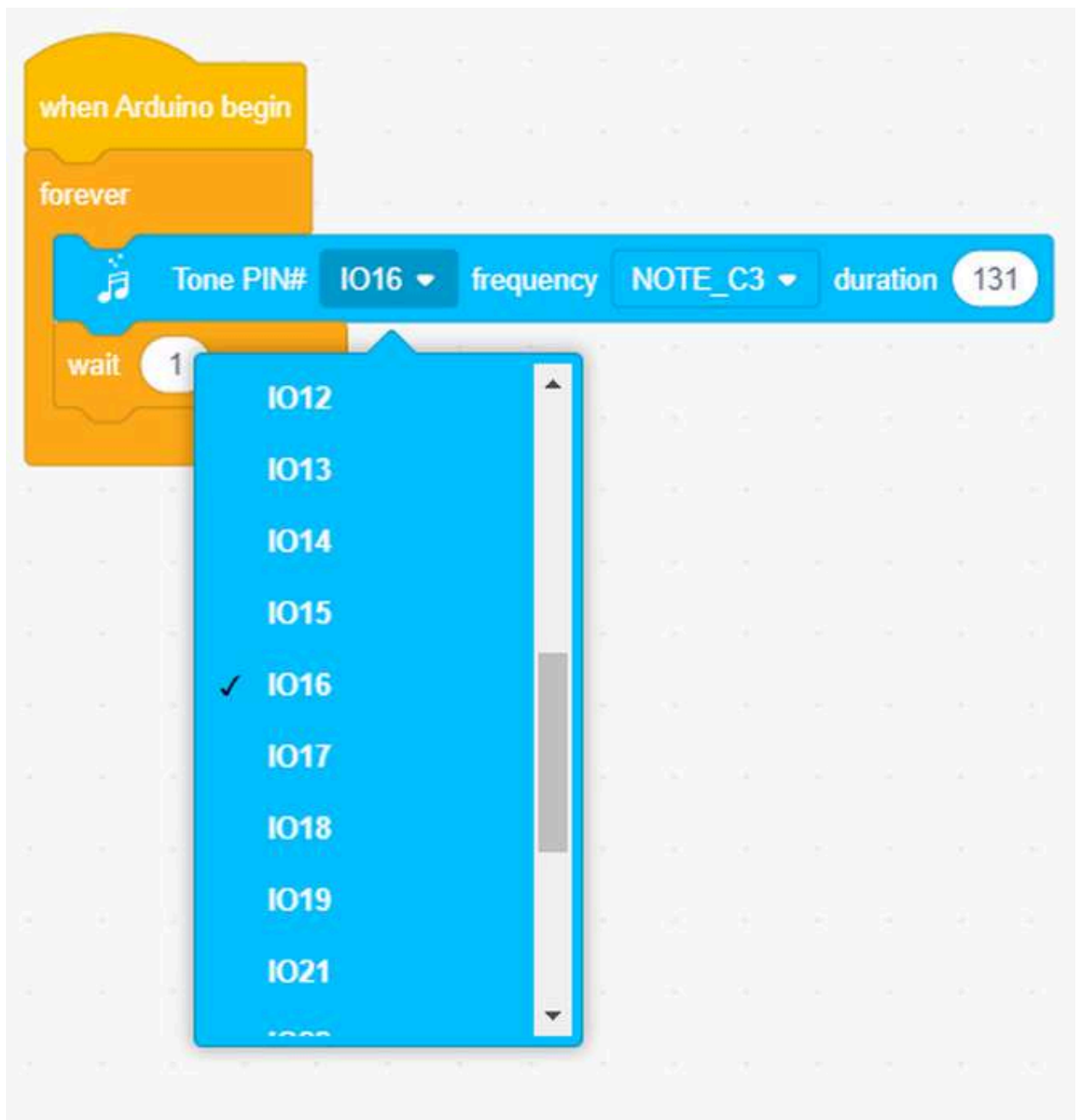
	1	2	3	4	5	6	7
A	221	248	278	294	330	371	416
B	248	278	294	330	371	416	467
C	131	147	165	175	196	221	248
D	147	165	175	196	221	248	278
E	165	175	196	221	248	278	312
F	175	196	221	234	262	294	330
G	196	221	234	262	294	330	371

	1	2	3	4	5	6	7
A	441	495	556	589	661	742	833
B	495	556	624	661	742	833	935
C	262	294	330	350	393	441	495
D	294	330	350	393	441	495	556
E	330	350	393	441	495	556	624
F	350	393	441	495	556	624	661
G	393	441	495	556	624	661	742

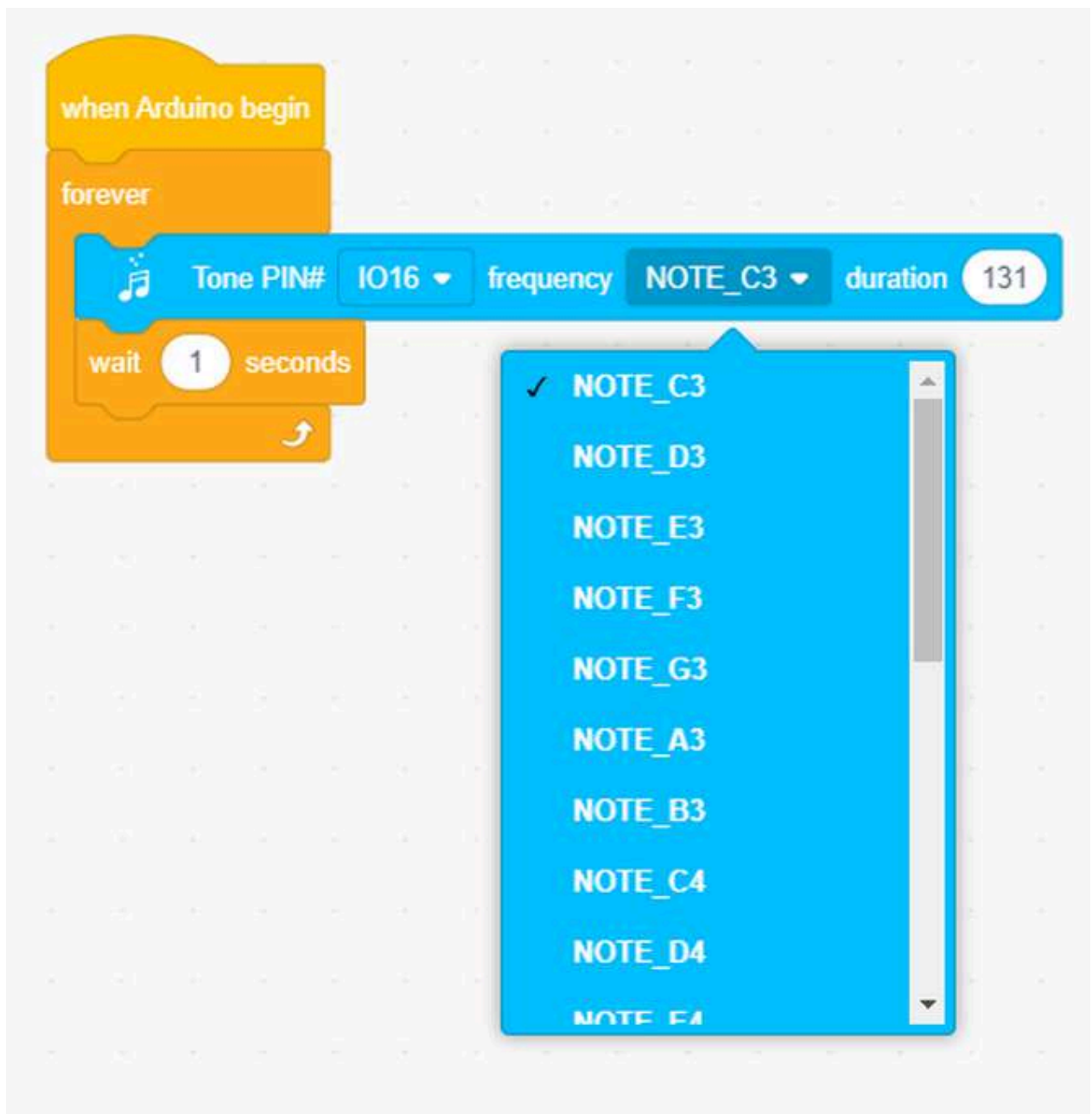
	1	2	3	4	5	6	7
A	882	990	1112	1178	1322	1484	1665
B	990	1112	1178	1322	1484	1665	1869
C	525	589	661	700	786	882	990
D	589	661	700	786	882	990	1112
E	661	700	786	882	990	1112	1248
F	700	786	882	935	1049	1178	1322
G	786	882	990	1049	1178	1322	1484

Cod:

- Trageți un bloc "Ton" din  așa cum se arată mai jos și setați pinul la IO16.



- Puteți selecta o frecvență după bunul plac.



- Fără ton: Este folosit pentru a opri toate tonurile.



Cod complet:



Rezultatul testului:

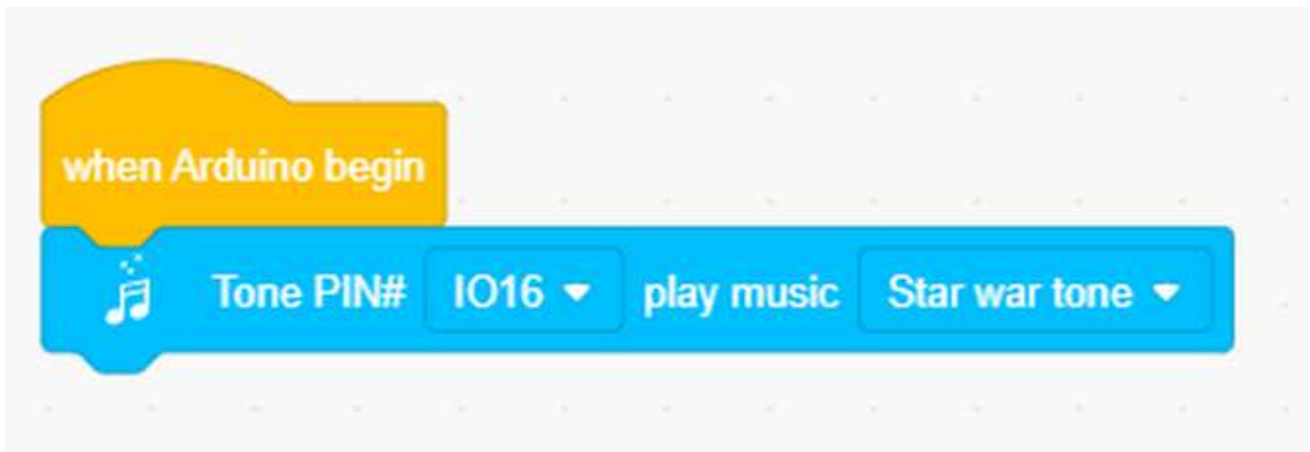
Metoda 1: Soneria continuă să emită sunet.

Metoda 2: Alarmer sonore prin funcția tone().

Expansiune: Redați muzică

Redați muzică prin tone().

Cod complet:



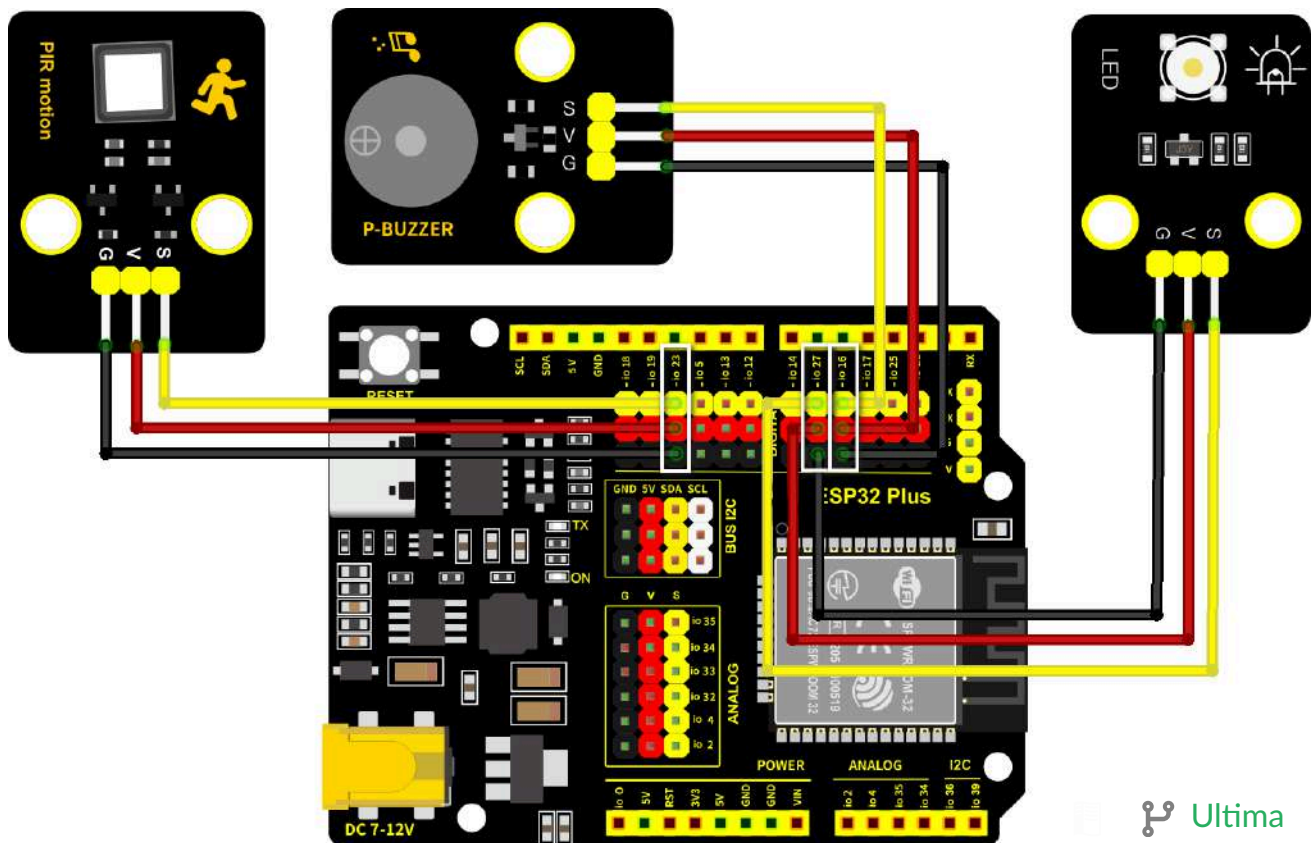
4.3.4 Sistem de alarmă

În acest experiment, vom construi un sistem de alarmă printr-o mișcare PIR senzor, un sonerie și un LED. Când senzorul detectează o mișcare, sonerie emite sunet și LED-ul clipește pentru a aminti de o invazie.

Schema:

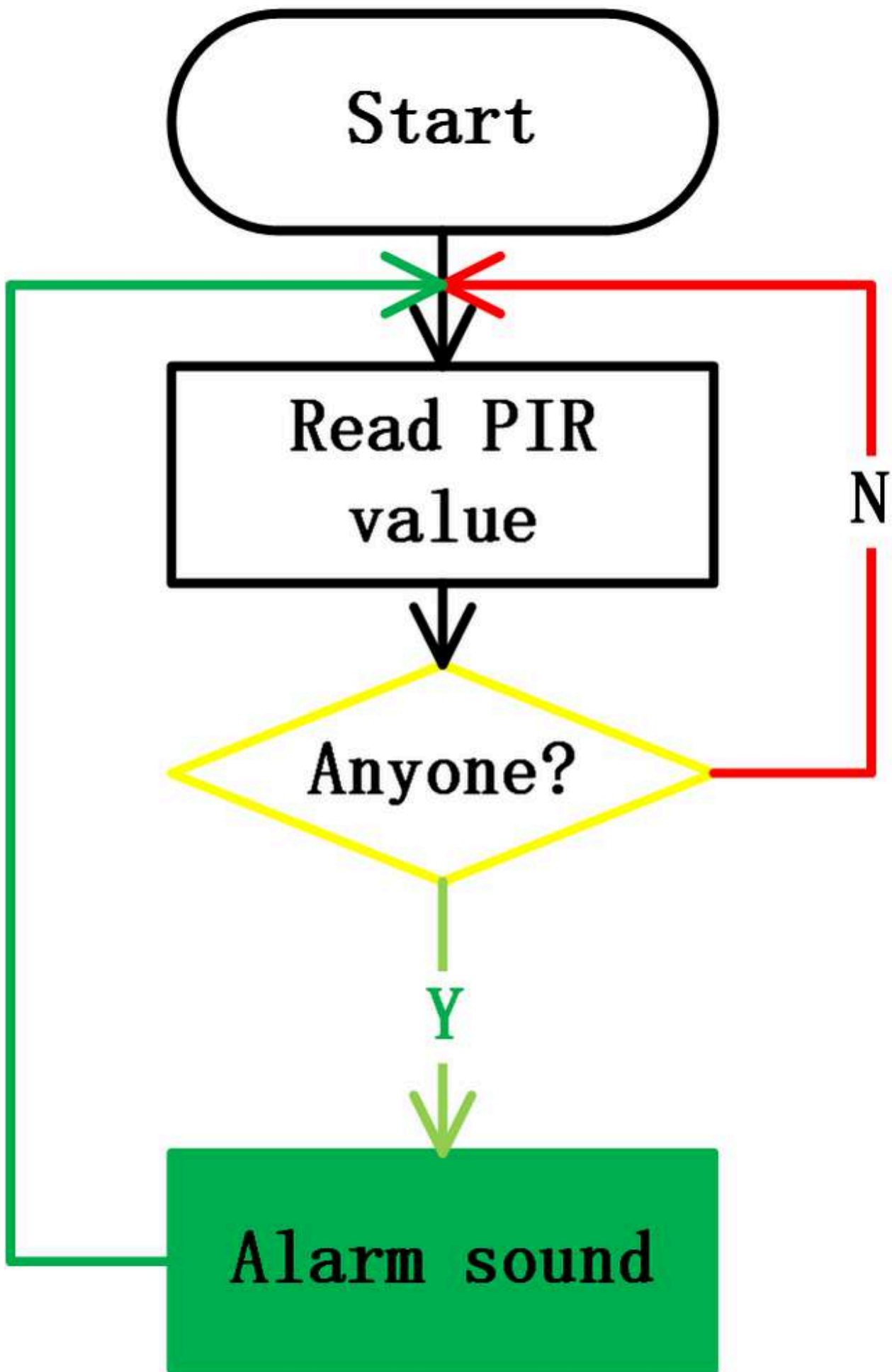
Conectați senzorul de mișcare PIR la io23, soneria la io16 și LED-ul la io27.

Atenție: Conectați galben la S (Semnal), roșu la V (Alimentare) și negru la GND. Nu le inversați!



Cod de testare:

Fluxul de cod:



Cod complet:

```

when Arduino begin
  serial 0 begin baudrate 115200
  forever
    if Read interface IO23 PIR motion sensor then
      serial 0 print Someone warp
      LED pin IO27 output HIGH
      Tone PIN# IO16 frequency NOTE_C3 duration 131
      Tone PIN# IO16 frequency NOTE_A4 duration 131
      Tone PIN# IO16 frequency NOTE_B4 duration 131
      Tone PIN# IO16 frequency NOTE_D5 duration 131
      Tone PIN# IO16 frequency NOTE_F5 duration 131
      LED pin IO27 output LOW
    else
      NoTone IO16
      serial 0 print No one warp
  
```

Rezultatul testului:

Încărcați codul și sistemul de alarmă începe să funcționeze. Când detectează un alarme de mișcare, sonerie și LED-uri clipesc.

4.3.5 ÎNTREBĂRI FRECVENTE

Î: Tonurile soneriei nu sunt exacte cu cele reale.

R: Acest buzzer obișnuit doar stimulează tonurile, deci nu este capabil să se întâlnească cerințe profesionale. Dacă doriți tonuri standard, un este necesar un vorbitor specializat.

Î: Senzorul de mișcare PIR dezinformează rezultatele.

R: Acest senzor de mișcare PIR nu este, de asemenea, unul profesional.

Vă rugăm să garantați următoarele situații pentru a evita o dezinformare:

- Evitați obiectele suflate de vânt să fluture în zona de detectare, cum ar fi perdele, îmbrăcăminte și flori.
- Evitați lumina puternică în zona de detectare, cum ar fi lumina soarelui, mașina lumini, spoturi și alte surse de lumină.
- Și așa mai departe...

4.4 Proiect: Sistem de detectare a ploii

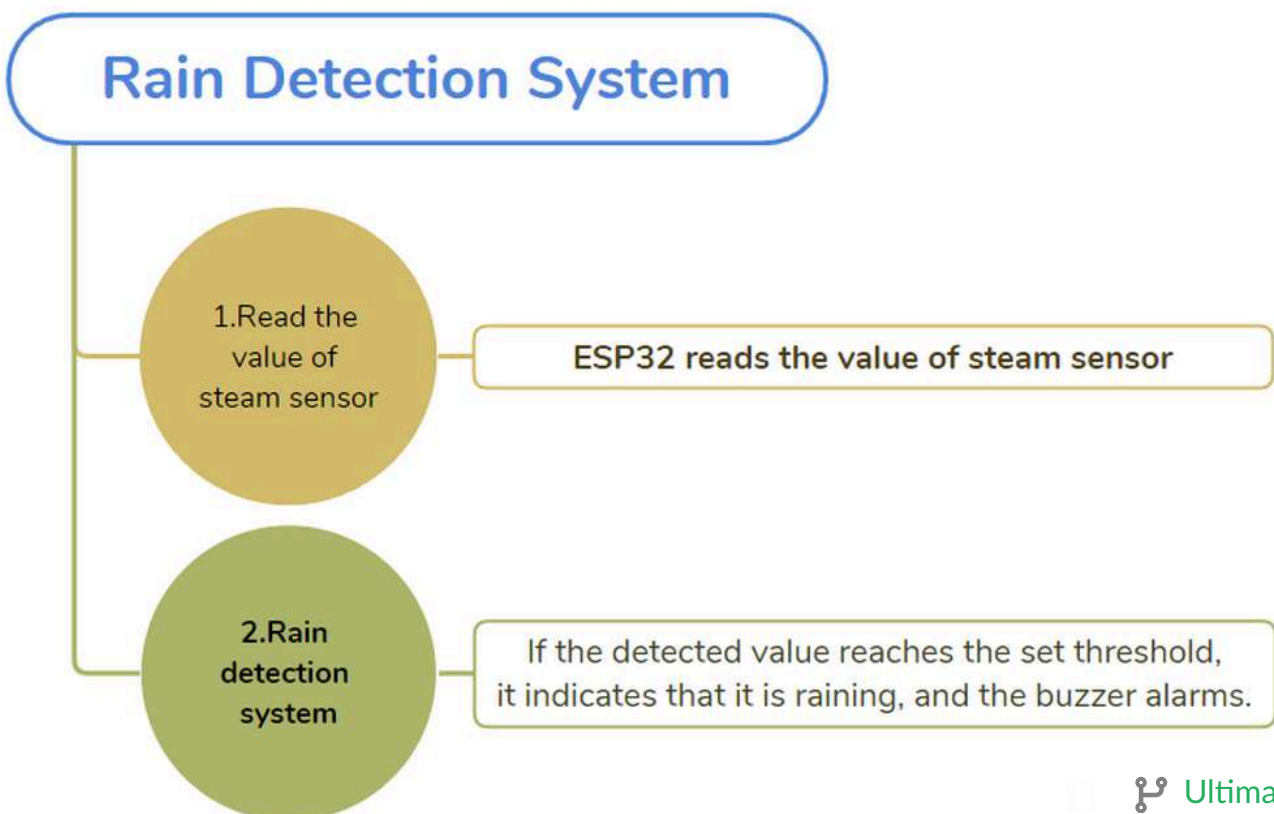
NOTĂ: Stropirea cu apă pe senzori (cu excepția senzorului de abur) poate provoca un scurtcircuit sau module să nu mai funcționeze. Dacă bateriile se udă, chiar și poate apărea o explozie. Fii foarte atent! Pentru utilizatorii mai tineri, vă rugăm operează cu părinții tăi. Pentru a garanta securitatea, vă rugăm să respectați instrucțiunile și reglementări de siguranță.

În acest proiect, vom crea un sistem de detectare a ploii printr-un abur senzor. Când este detectată ploua, ESP32 declanșează diverse acțiuni, cum ar fi trimiterea de mesaje, activarea aspersoarelor și aprinderea luminilor. Prin acest sistem, cantitatea de precipitații poate fi monitorizată, iar scurgerile de apă pot fi să fie detectate și pe acoperișuri sau în clădiri.

În plus, este ușor să conectați senzorul de abur la placa ESP32, care formează un sistem simplu, dar eficient de detectare a ploii.



4.4.1 Diagrama de flux



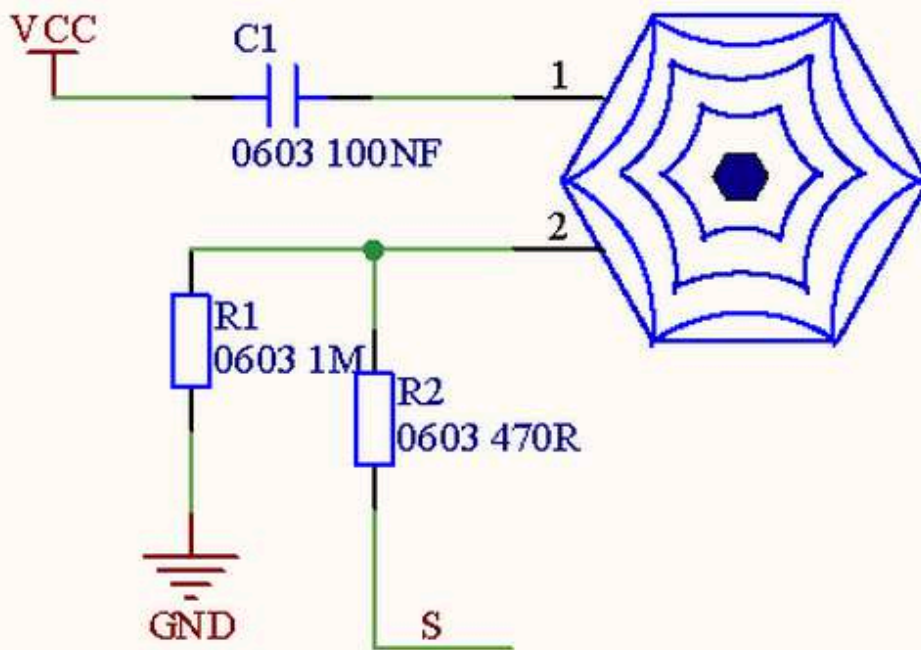
4.4.2 Senzor de abur

Descriere:

Senzorul de abur detectează prezența apei, deci este de obicei utilizat în detectarea ploii. Dacă ploaia lovește tamponul conductiv de pe senzor, acesta va trimite un semnal către placa Arduino.



Diagramă schematică:



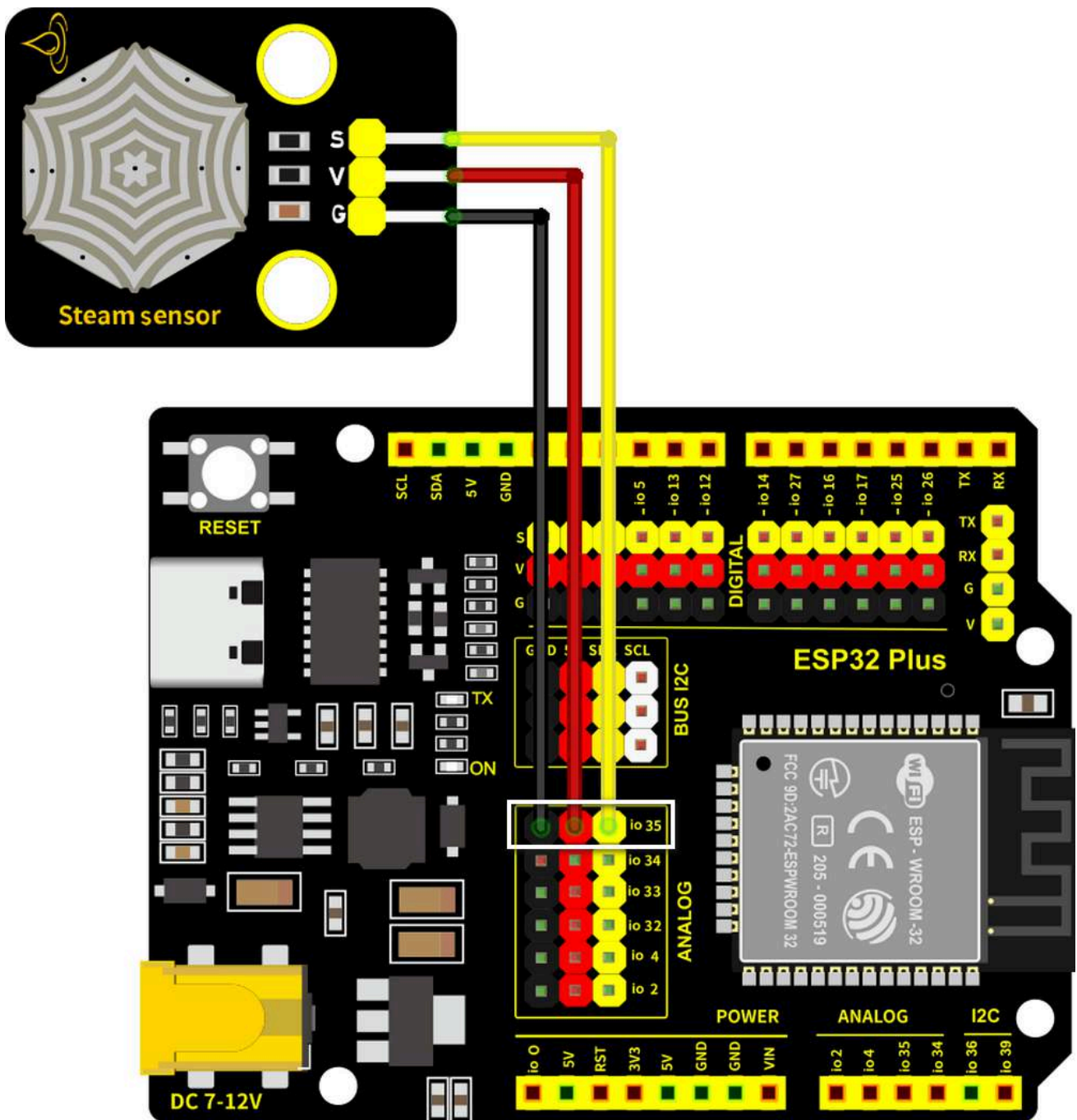
Parametrii:

- Tensiune: 3 ~ 5V
- Curent: 1.5mA
- Putere: 7.5mW

Schema:

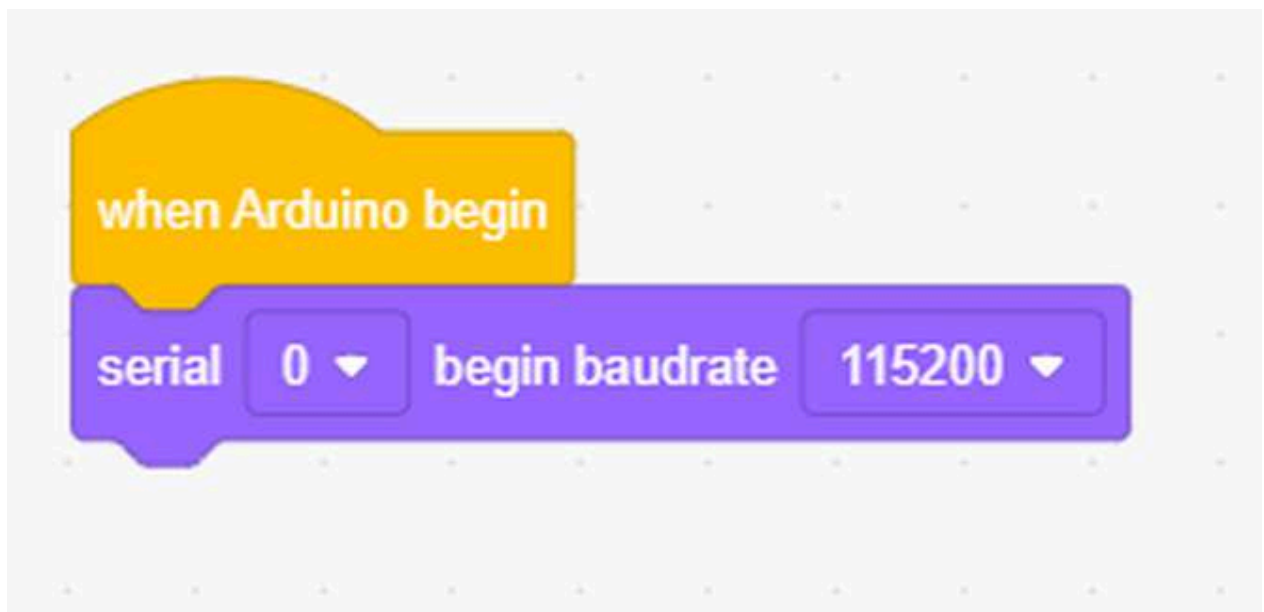
Conectați senzorul de abur la io35.

Atenție: Conectați galben la S (Semnal), roșu la V (Alimentare) și negru la GND. Nu le inversați!

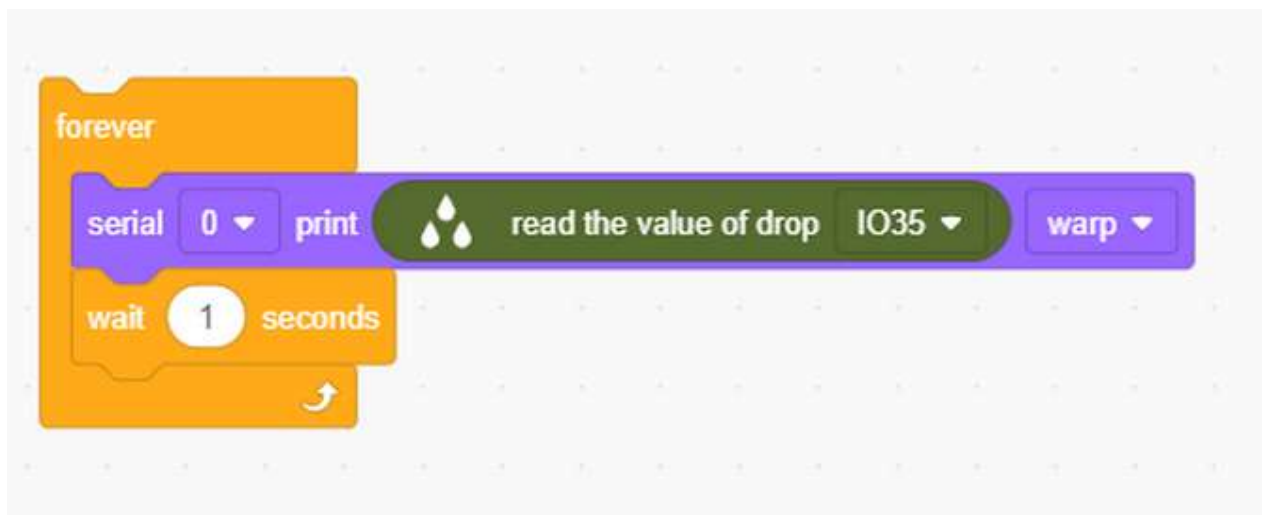


Cod de testare:

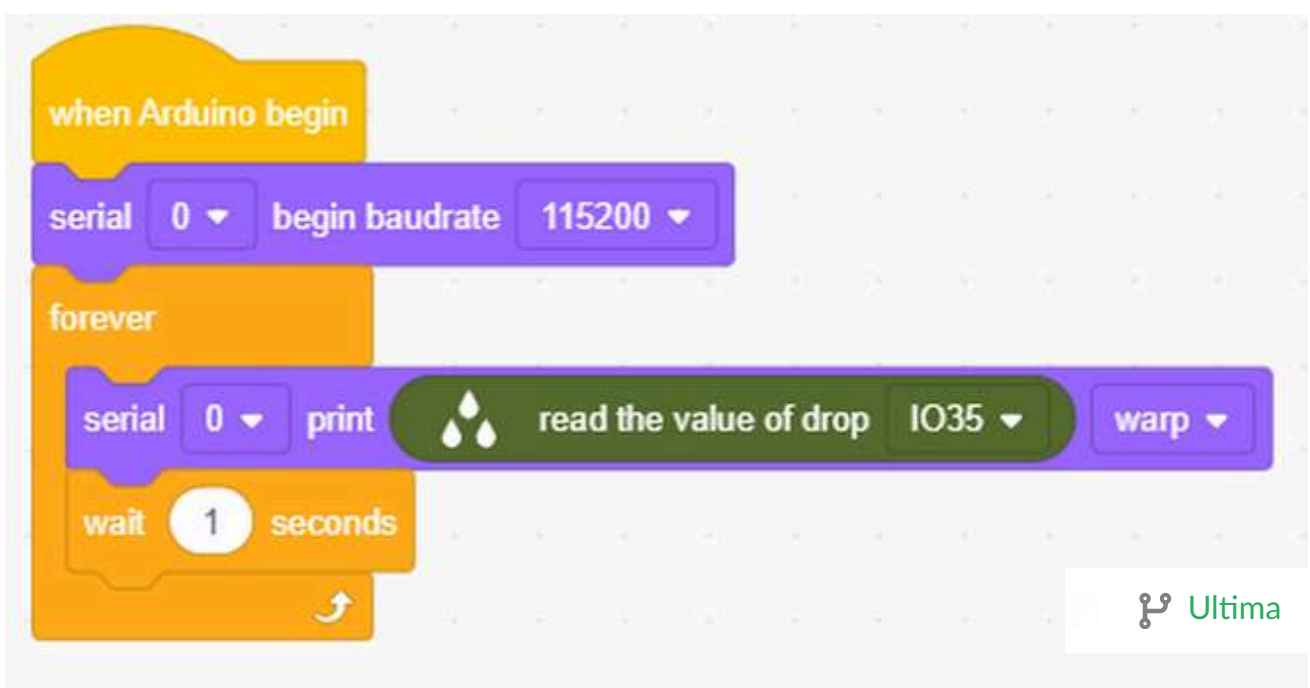
- Inițializați portul serial.



- Citiți valoarea senzorului la pinul io35 și imprimați-o pe secundă.



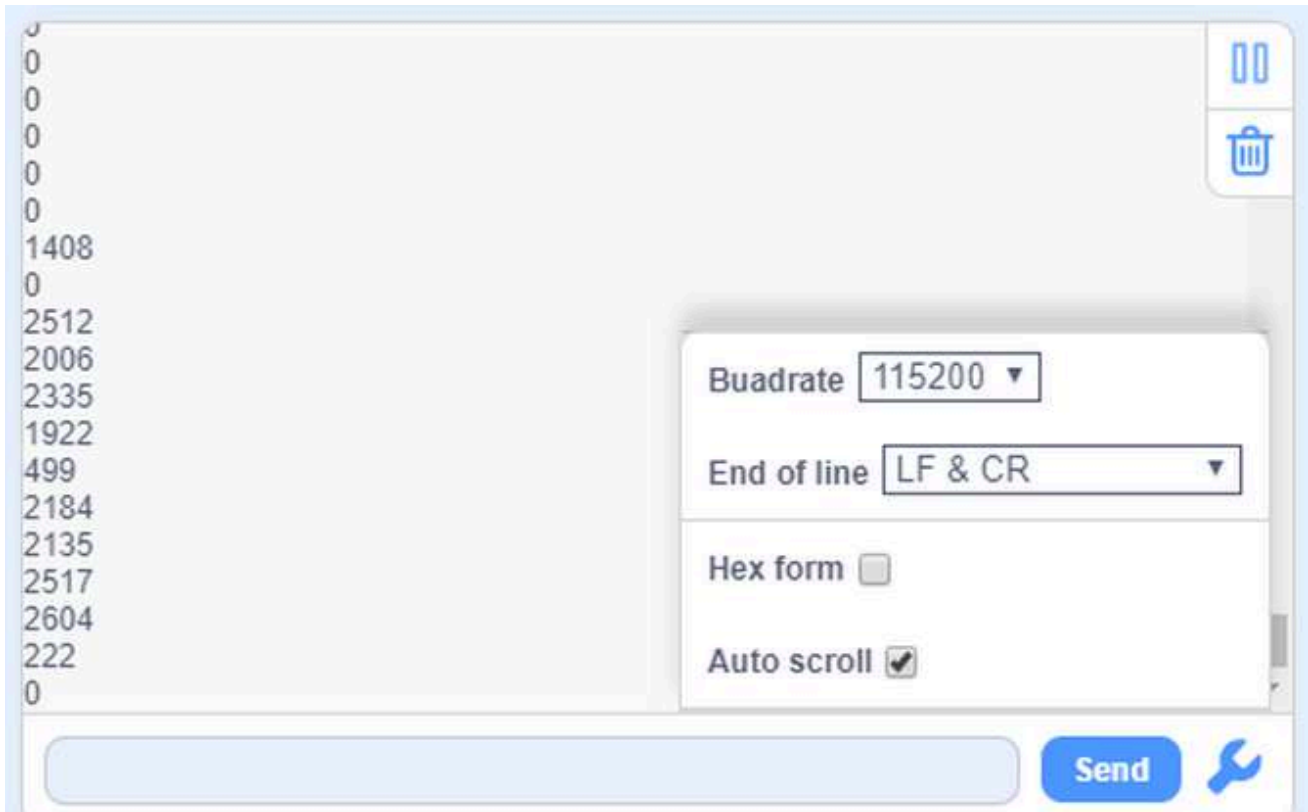
Cod complet:



Rezultatul testului:

Atingeți zona de detectare cu un deget umed. Cu cât suprafața este mai mare atins este, cu atât valoarea va fi mai mare.

Puteți deschide monitorul serial pentru a observa valoarea detectată în prezent (interval: 0~4095).



4.4.3 Sistem de detectare a ploii

Descriere:

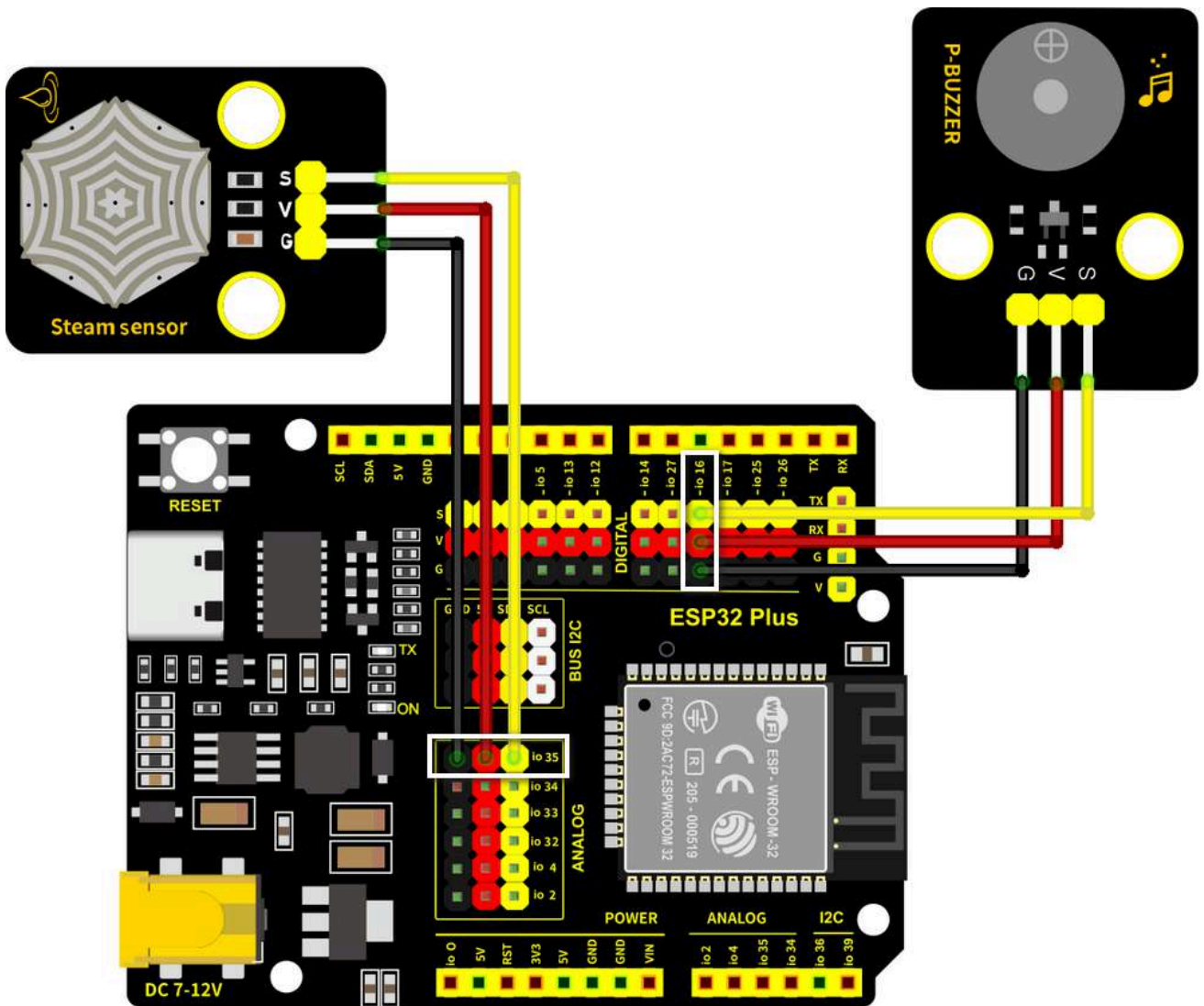
Când senzorul de abur detectează ploaie, trimite un semnal către placă pentru declanșează diverse acțiuni, de exemplu, alarmele sonore pentru a reaminti că Plouă. Acest lucru este util în special pentru grădinăritul în aer liber și agricol, permițând utilizatorilor să ia măsurile de precauție necesare pentru a evita udare excesivă.

În plus, acest sistem poate fi utilizat pentru a detecta scurgerile de apă pentru a preveni daune cauzate de pătrunderea apei. În general, senzorul de abur este versatil și Eficient în diverse aplicații.

Schema:

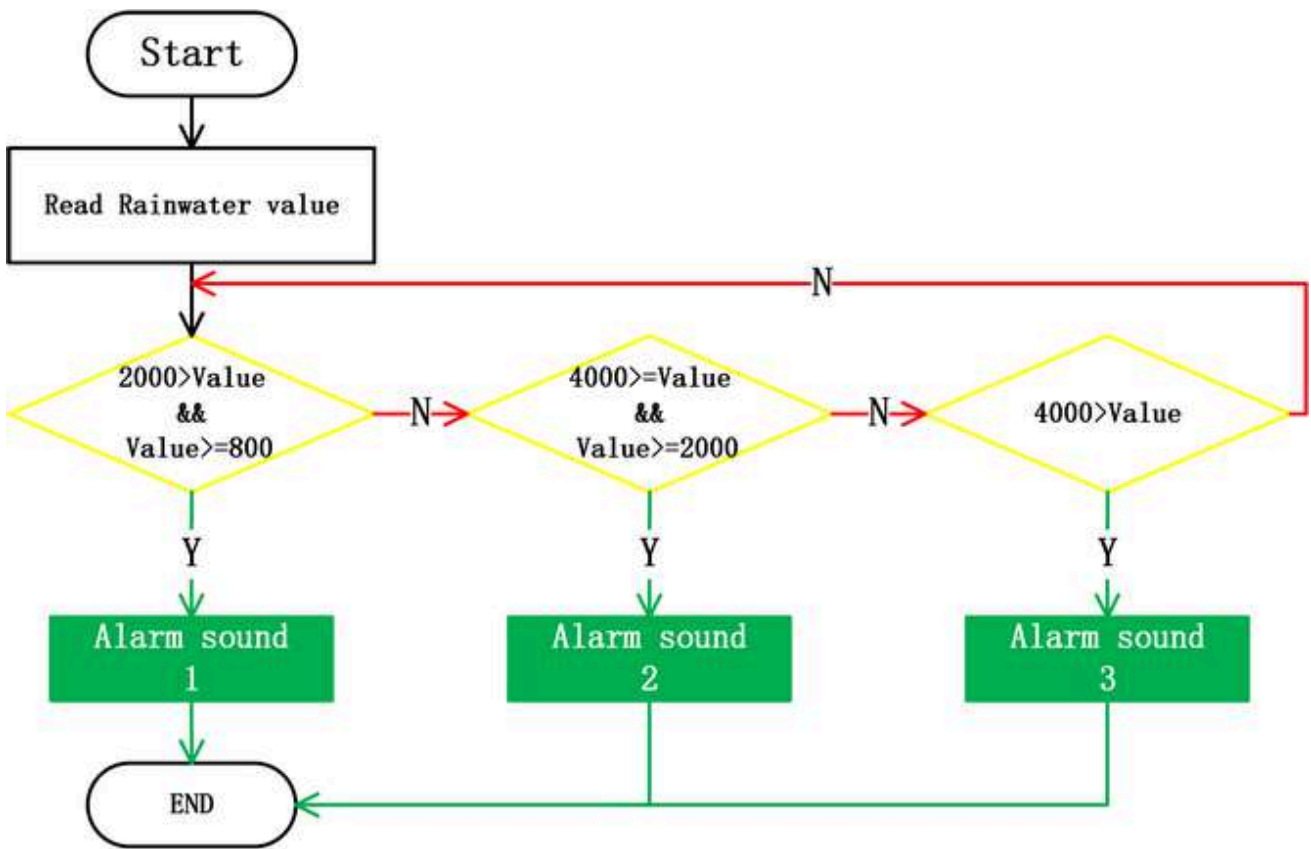
Conectați senzorul de abur la io35 și soneria la io16.

Atenție: Conectați galben la S (Semnal), roșu la V (Alimentare) și negru la GND. Nu le inversați!



Cod de testare:

Fluxul de cod:



Cod:

- Inițializați portul serial și definiți un **element** variabil ca valoarea senzorului primit.

```

when Arduino begin
  serial 0 begin baudrate 115200
  Declare Global variable Type int Name item Assigned to 0
  
```

- Primiți valoarea senzorului și imprimați-o pe monitorul serial.

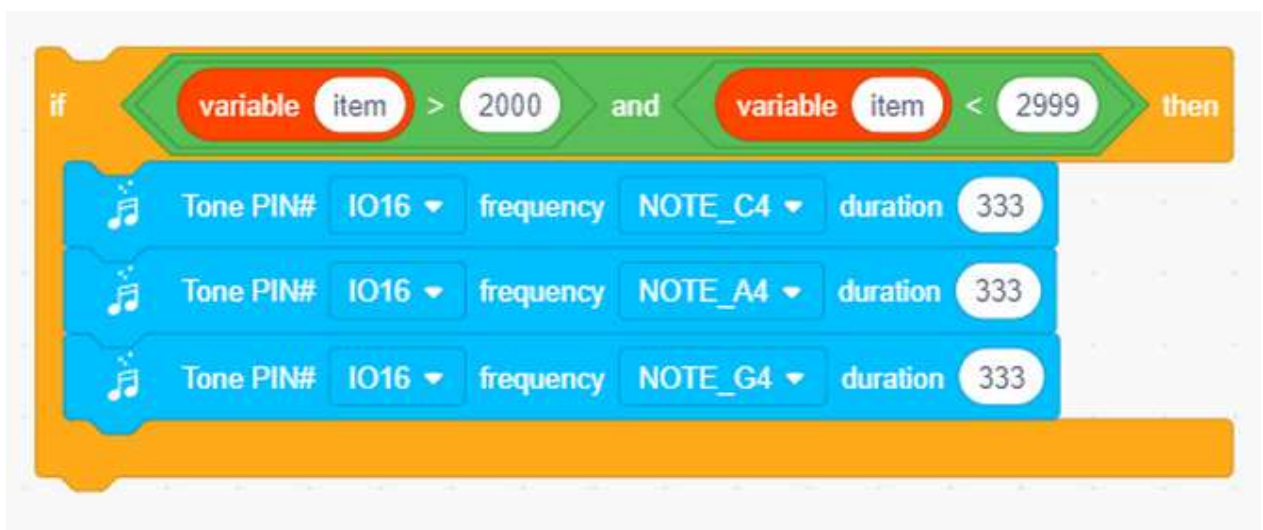
```

forever
  Set item variable by read the value of drop IO35
  serial 0 print variable item warp
  
```

- Valoarea recepționată detectată de senzor este cuprinsă între 800 ~ 1999:



- Valoarea primită detectată de senzor este în limita a 2000 ~ 2999:



- Valoarea primită detectată de senzor este mai mare de 3000:



- La sfârșitul blocurilor de cod, adăugați un "Fără ton" pentru a dezactiva Sonerie.



Cod complet:

when Arduino begin

serial 0 begin baudrate 115200

Declare Global variable Type int Name item Assigned to 0

forever

Set item variable by read the value of drop IO35

serial 0 print variable item warp

if variable item > 800 and variable item < 1999 then

Tone PIN# IO16 frequency NOTE_C3 duration 555

Tone PIN# IO16 frequency NOTE_D3 duration 555

Tone PIN# IO16 frequency NOTE_E3 duration 555

if variable item > 2000 and variable item < 2999 then

Tone PIN# IO16 frequency NOTE_C4 duration 333

Tone PIN# IO16 frequency NOTE_A4 duration 333

Tone PIN# IO16 frequency NOTE_G4 duration 333

if variable item > 3000 then

Tone PIN# IO16 frequency NOTE_G5 duration 111

Tone PIN# IO16 frequency NOTE_A5 duration 111

Tone PIN# IO16 frequency NOTE_D5 duration 111

NoTone IO16

Rezultatul testului:

Cu cât valoarea detectată este mai mare, cu atât încărcătorul sunetul emis de sonerie va fi.

4.4.4 ÎNTREBĂRI FRECVENTE

Î: Senzorul de abur este rezistent la apă?

R: Zona de detectare poate fi expusă la apă, dar joncțiunile de sârmă nu sunt rezistente la apă. În timpul experimentului, vă rugăm să acordați atenție cantitatea de apă să nu fie prea mare pentru a preveni scurtcircuitul.

Î: Deși a trecut mult timp de când senzorul a detectat apă, soneria continuă să bâzâie.

R: Continuă să bâzâie pentru că există încă pete de apă în zona de detectare. Vă rog doar curățați-l.

4.5 Proiect: Sistem de energie solară

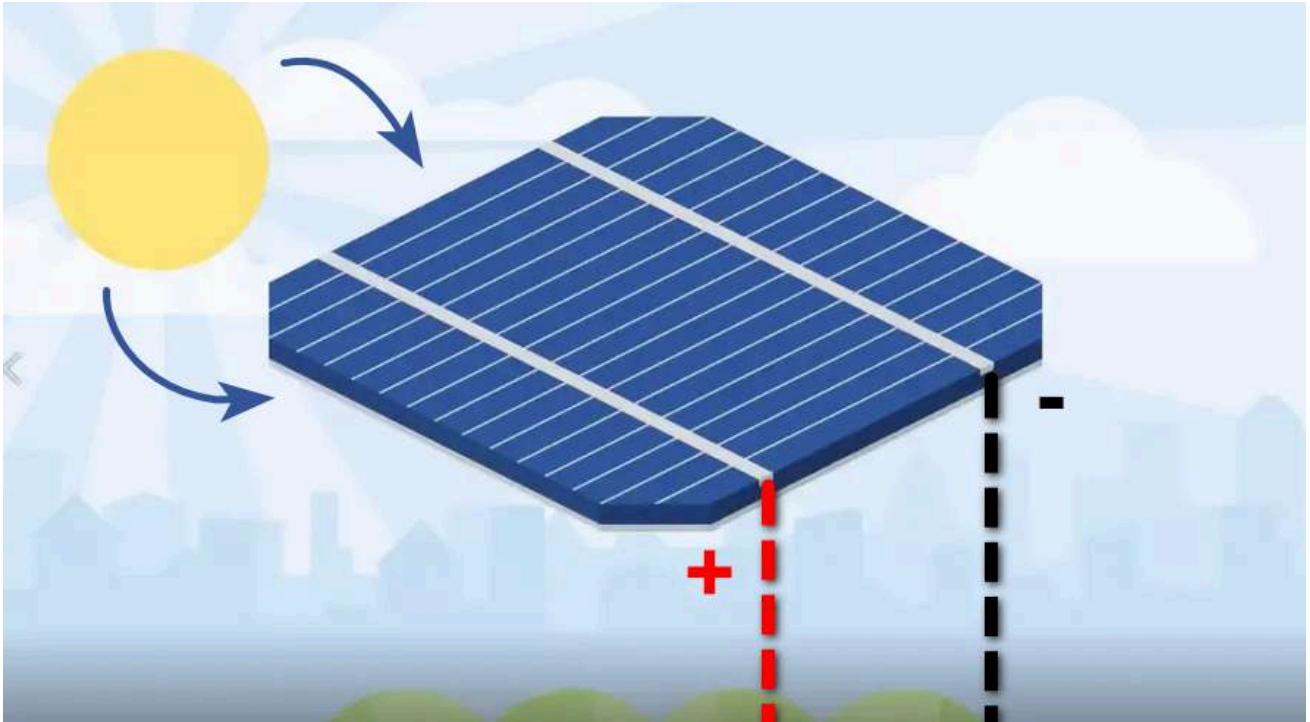


4.5.1 Descriere

Panoul solar transformă energia solară în electricitate pentru LED. Este potrivit pentru aplicații multiple, cum ar fi iluminatul exterior, dispozitive care se încarcă și fac backup de energie. Prin urmare, puteți stabili un sistem de energie solară sofisticat și eficient de propriul dvs. nevoi.

4.5.2 Principiul de funcționare

Cum transformă panoul solar energia solară în electricitate?



Panoul solar absoarbe lumina și transformă direct sau indirect energia solară radiații în electricitate. Comparativ cu energia obișnuită pe cărbune energia solară, eoliană și hidroenergie sunt mai economice și Prietenoase cu mediul.

Cum se transformă lumina în electricitate?

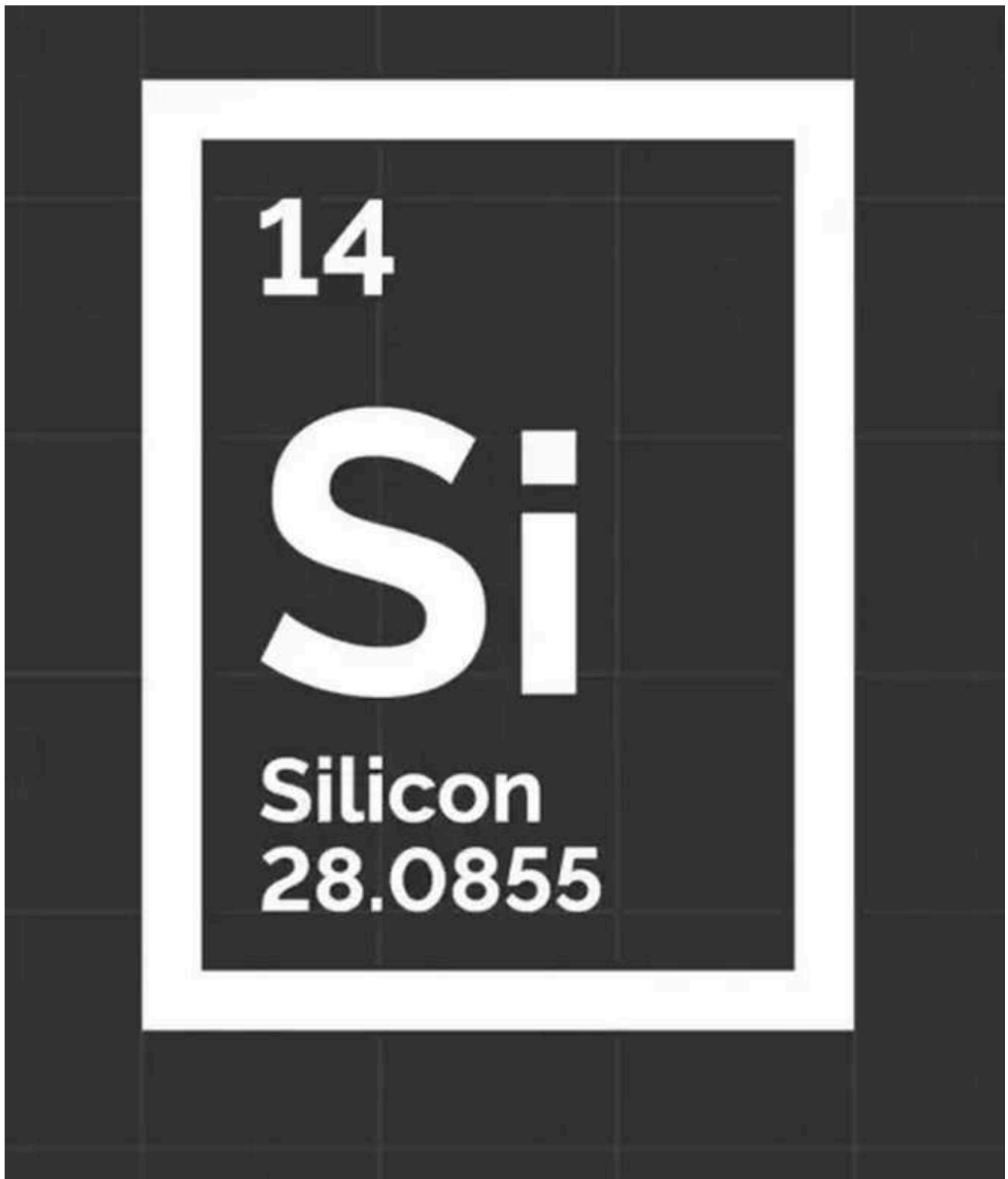
În continuare, să vorbim despre procesul de conversie din interior în exterior un panou solar.

Soarele emite energie în unde cu o gamă largă de lungimi de undă, de la de la ultraviolet la vizibil la lumină infraroșie.

- Lungimea de undă a ultravioletului: 150 ~ 400nm;
- Lungimea de undă a luminii vizibile: 400 ~ 760nm;
- Lungimea de undă a luminii infraroșii: 760 ~ 4000nm;

Panoul absoarbe una dintre aceste game de lungimi de undă și le convertește în electricitate. Dar cum? Să mergem mai departe.

Partea activă a majorității celulelor de panouri solare este realizată dintr-un semiconductor – siliciu (Si).

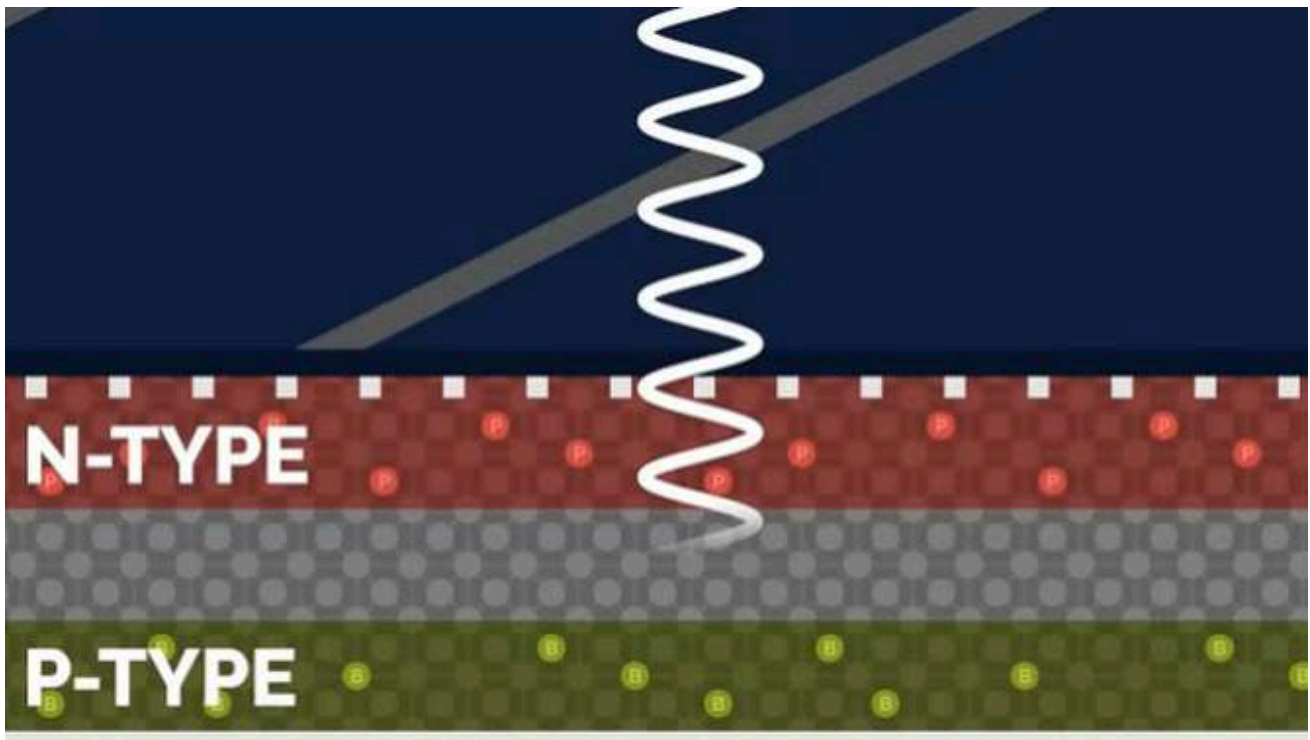


Conductivitatea unui semiconductor este între un conductor și un izolator la temperatura atmosferică. În general, nu poate conduce bine, cu toate acestea, conductivitatea sa se îmbunătățește în anumite condiții.



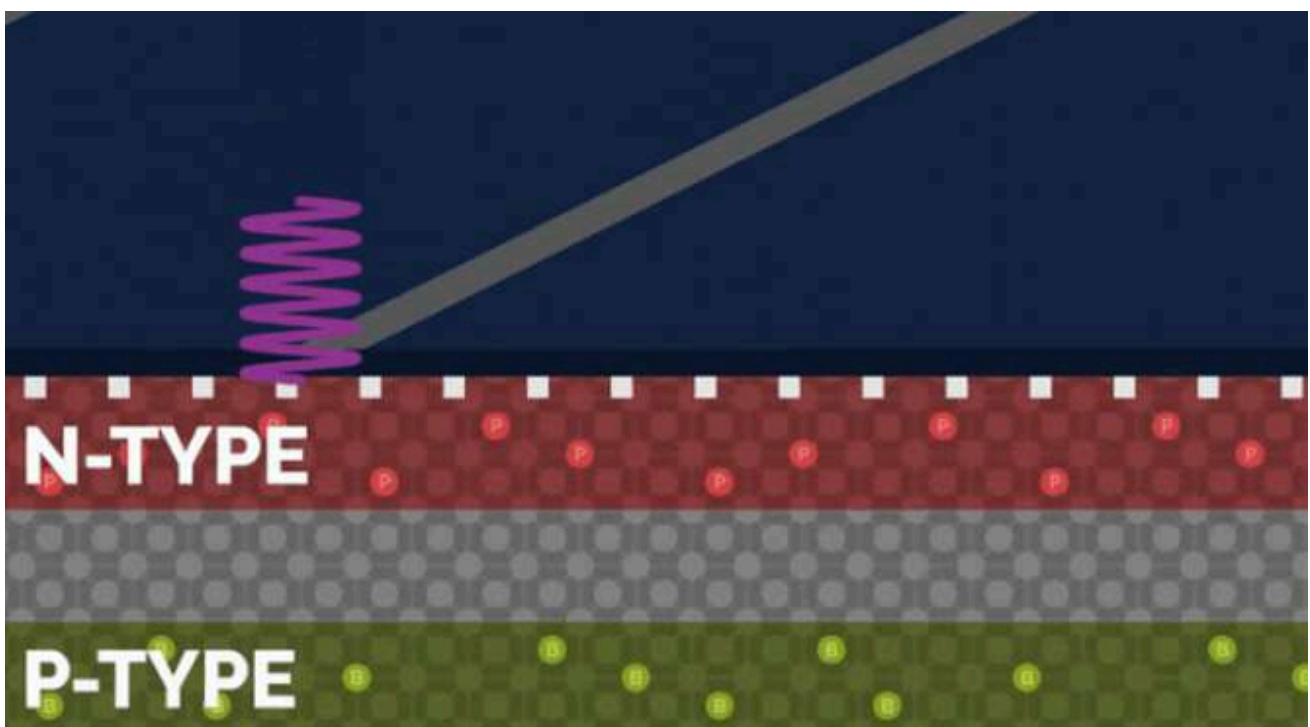
Diagrama de mai sus arată structura internă a semiconductorului în celulă solară, care este împărțită în trei straturi:

1. **Stratul superior (partea roșie)** este format din siliciu (Si) și puțin Fosfor (P). Cea din urmă poartă mai mulți electroni decât prima, furnizând suficienți electroni pentru stratul superior. Datorită acestor electroni care se mișcă liber, acest strat este conductiv, deci se numește **negativ sau de tip N**.
2. **Stratul de mijloc (partea gri)** conține prea puțini electroni pentru a conduce.
3. **Stratul inferior (partea verde)** include în principal siliciu (Si) și Bor (B). Cea din urmă poartă mai puțini electroni decât prima, astfel încât un rareori puțini electroni se mișcă liber, provocând lipsa electronilor care sunt descrise ca sarcină pozitivă efectivă. Prin urmare, acest stratul este numit **Pozitiv sau de tip P**.

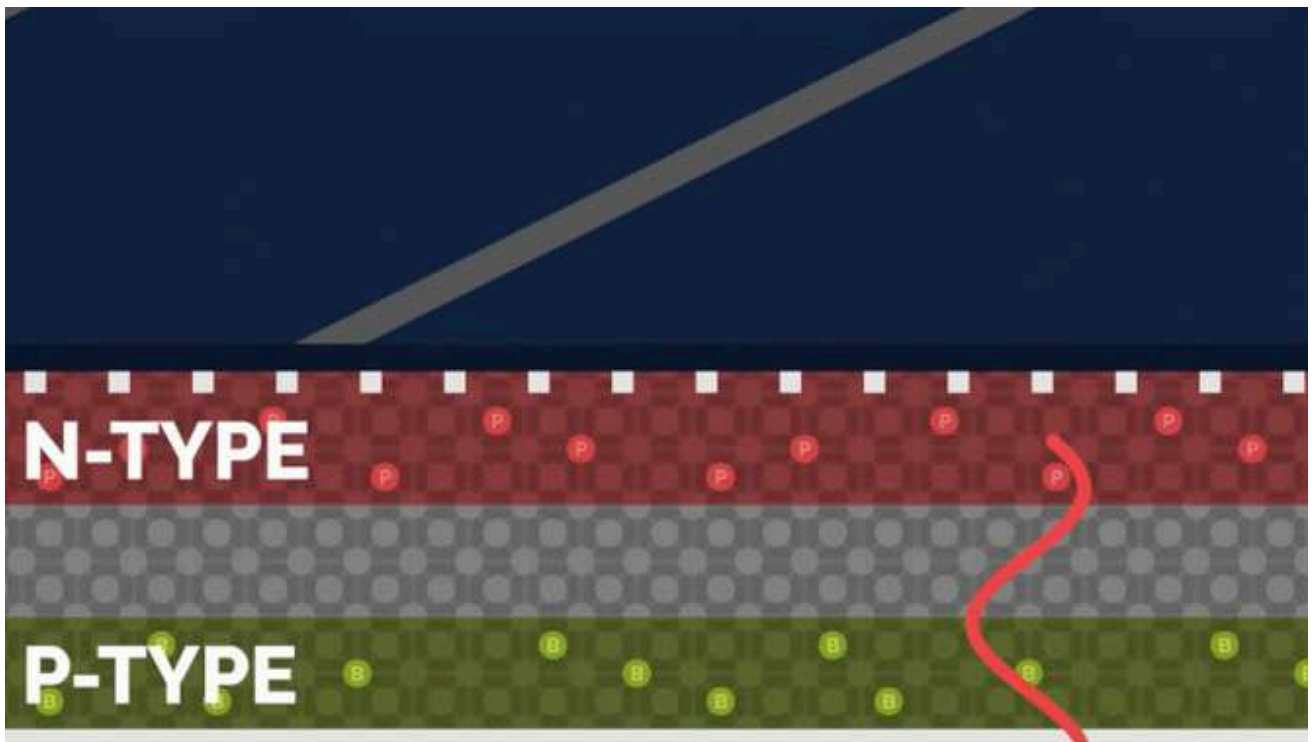


De obicei, doar stratul mijlociu al panoului solar absoarbe undele de lumină cu lungime de undă de 350 ~ 1140nm. În funcție de distribuția spectrului În paragrafele anterioare, absorbțiile sunt ultraviolete cu unde lungi, scurte unde infraroșu și lumină vizibilă.

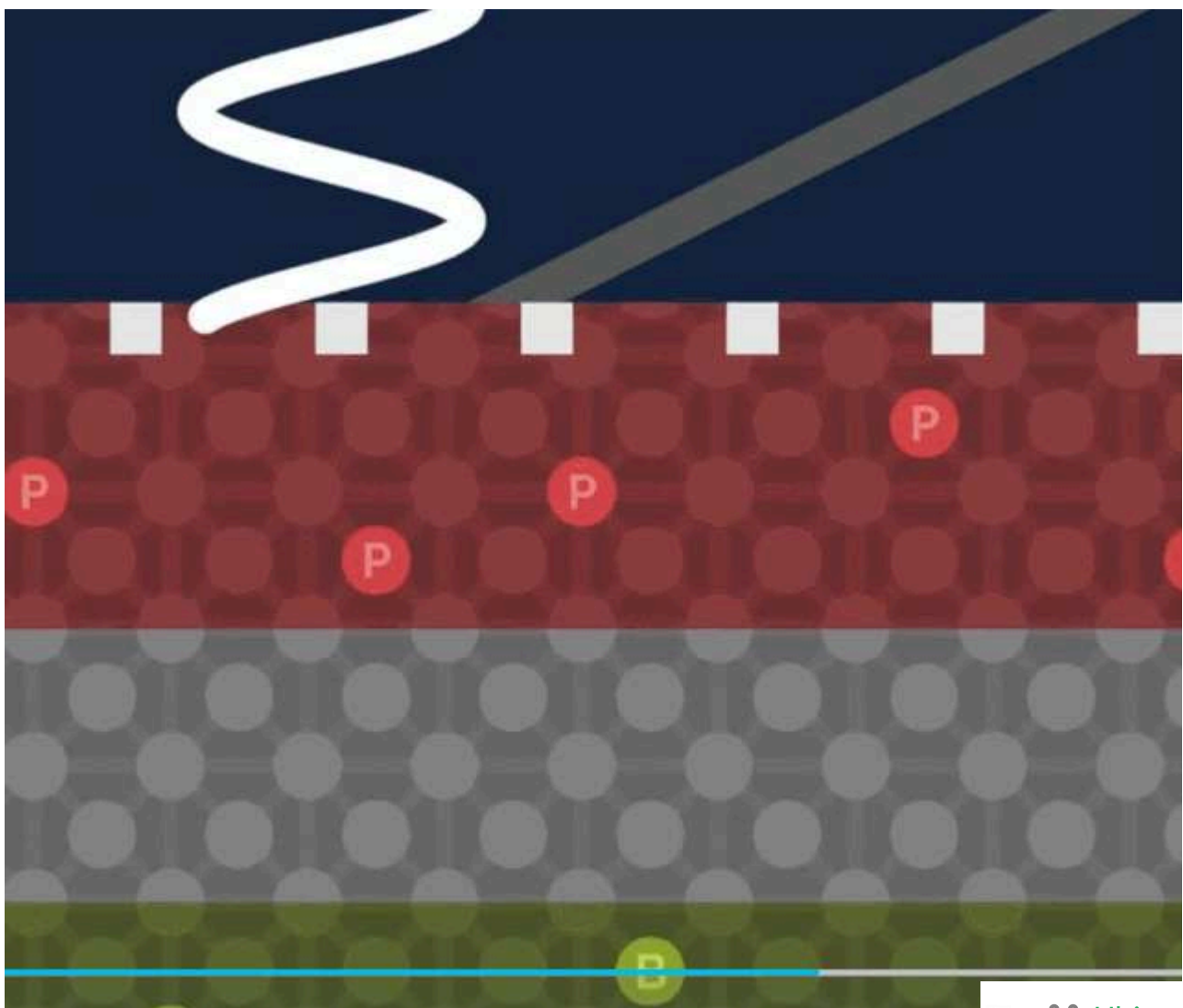
Lungimea de undă a ultravioletelor este atât de scurtă încât se oprește pe suprafață.



Lungimea de undă a luminii infraroșii este prea lungă pentru a fi absorbită de panou, deci de obicei trec sau sunt reflectate înapoi.



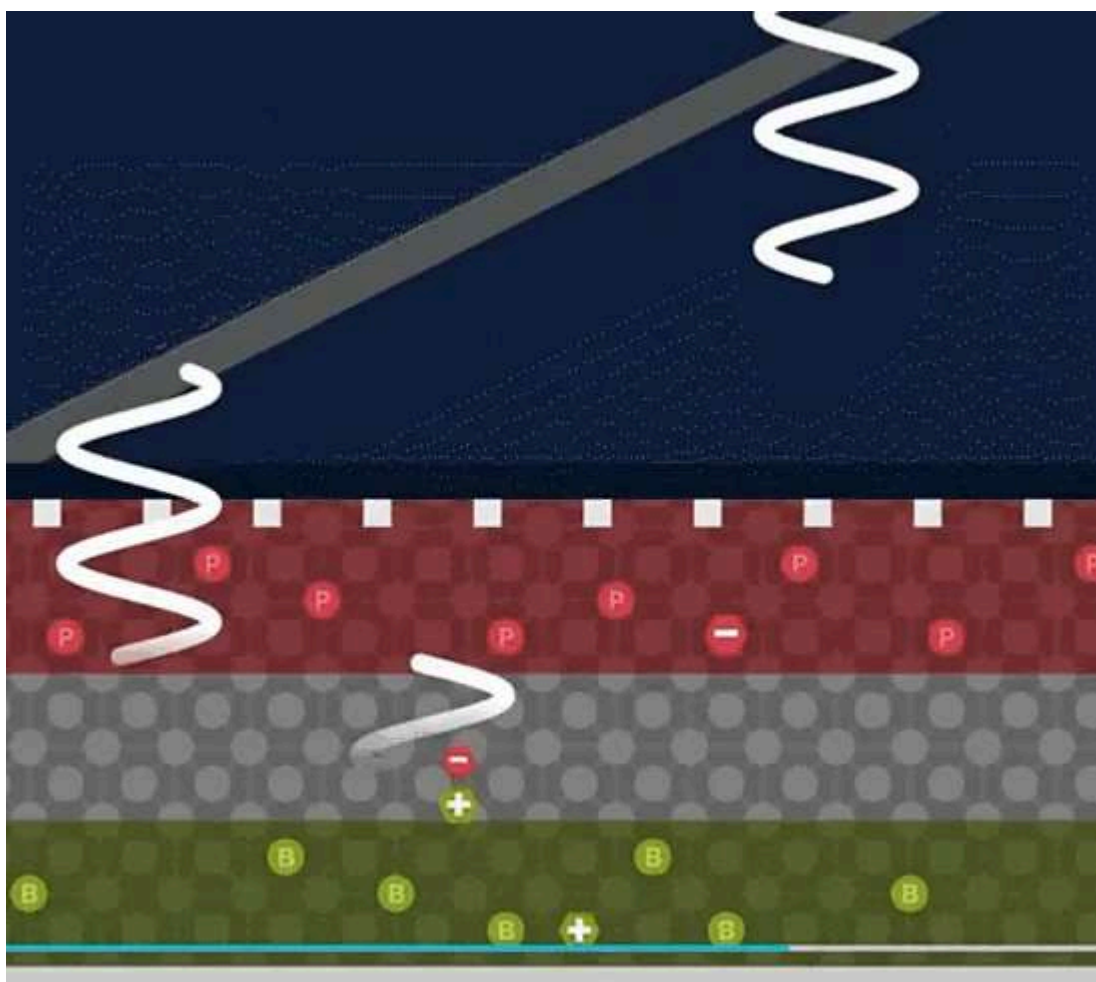
Stratul mijlociu absoarbe lumina și elimină electronii din siliciu în stratul superior, lăsându-le într-o stare liberă și găuri de electroni goale sunt generate în locul în care se aflau înainte.

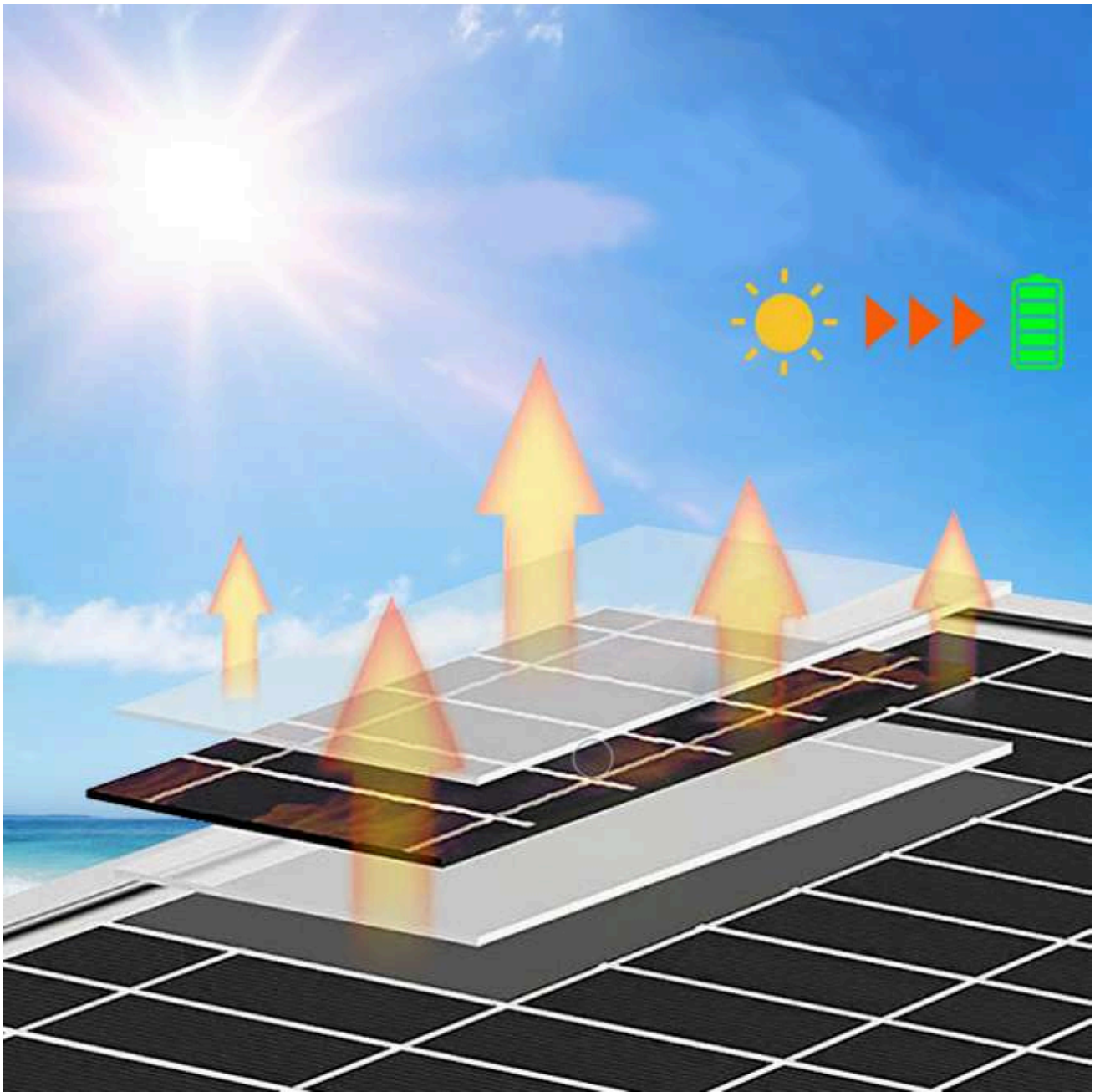


Găurile poartă o sarcină pozitivă. Între timp, electronii liberi se mișcă în sus pentru a ajunge la stratul de tip N, în timp ce găurile se mișcă în jos pentru a ajunge Strat de tip P.

În concluzie, electronii din straturile superioare și inferioare sunt eliminați după stratul intermediar care absoarbe energia solară. Prin urmare, stratul de tip N poartă sarcină negativă ca pol negativ, în timp ce stratul de tip P este încărcat pozitiv ca un pol pozitiv. În acest caz, atâta timp cât cele două straturile sunt conectate, conduce.

Dacă lumina soarelui strălucește pe panoul solar, situația de mai sus va dura, și va fi produs un număr mare de electroni liberi și găuri. Ca și concluzia este că electronii se mișcă în sus, în timp ce găurile se mișcă în jos, care formează cei doi poli și generează curent.





Energia solară este o sursă alternativă de energie, care prezintă durabilitate și rentabilitate.

Cu toate acestea, electricitatea generată de un panou solar poate fi convertită în câțiva wați de putere, ceea ce este suficient pentru un calculator sau un încărcător de telefon mobil, dar nu suficient pentru a rula un prăjitor de pâine de un kilowatt.

Sistemele de energie solară satisfac nevoile diferiților utilizatori și beneficiază de și mediul. Combinat cu programarea Arduino, acest tip de construcție o varietate de aplicații solare utile și eficiente, cum ar fi iluminat automat, încărcătoare și case inteligente.

În general, energia solară promite bine pentru o viitor durabil.

4.5.3 Parametri

- Tensiune: 5V
- Curent: 80mA
- Putere: 400mW

- Dimensiuni: 60 * 60mm

4.5.4 Rezultatul testului

Codurile nu sunt necesare în acest proiect. Important, aflăm despre Noua energie de mediu – energia solară.

Când este asigurată o iluminare bună, LED-ul se va aprinde în galben. Cel lumina este mai strălucitoare, cu atât LED-ul va fi mai strălucitor.

4.5.5 ÎNTREBĂRI FRECVENTE

Î: De ce panoul solar funcționează în continuare fără lumina soarelui?

R: Funcționează nu numai cu lumina soarelui, ci și cu lumina ambientală. Cu cât mai luminos lumina este, cu atât tensiunea va fi mai mare și cu atât LED-ul este mai ușor va fi.

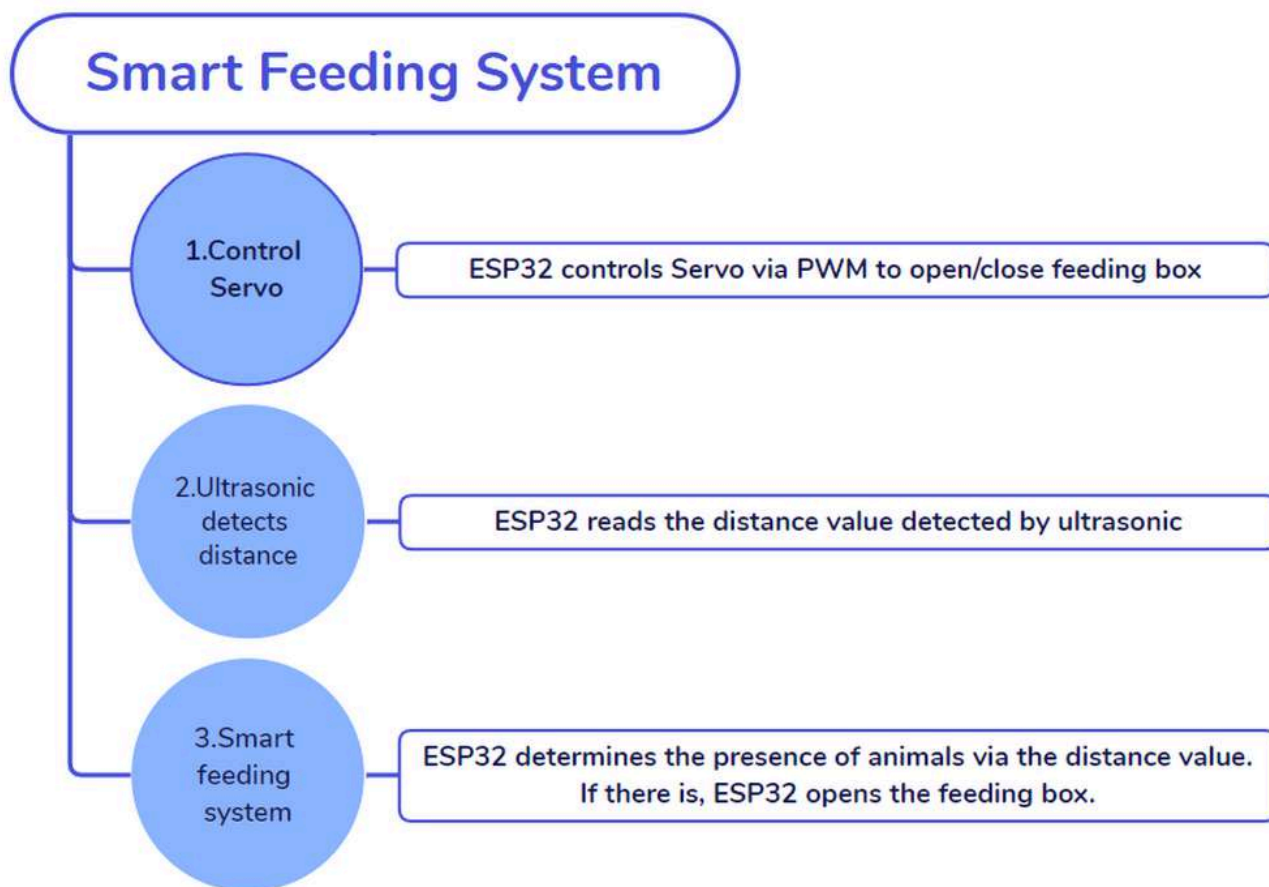
4.6 Proiect: Sistem inteligent de hrănire

În acest proiect, modulul cu ultrasunete detectează dacă animalele sunt în zona de alimentare, iar servo deschide automat cutia de alimentare pentru păsări. În plus, încorporarea IOT permite monitorizarea de la distanță a unor astfel de sisteme de alimentare care oferă multă comoditate.

În general, automatizarea și operarea de la distanță optimizează alimentarea pentru acest sistem.



4.6.1 Diagrama de flux



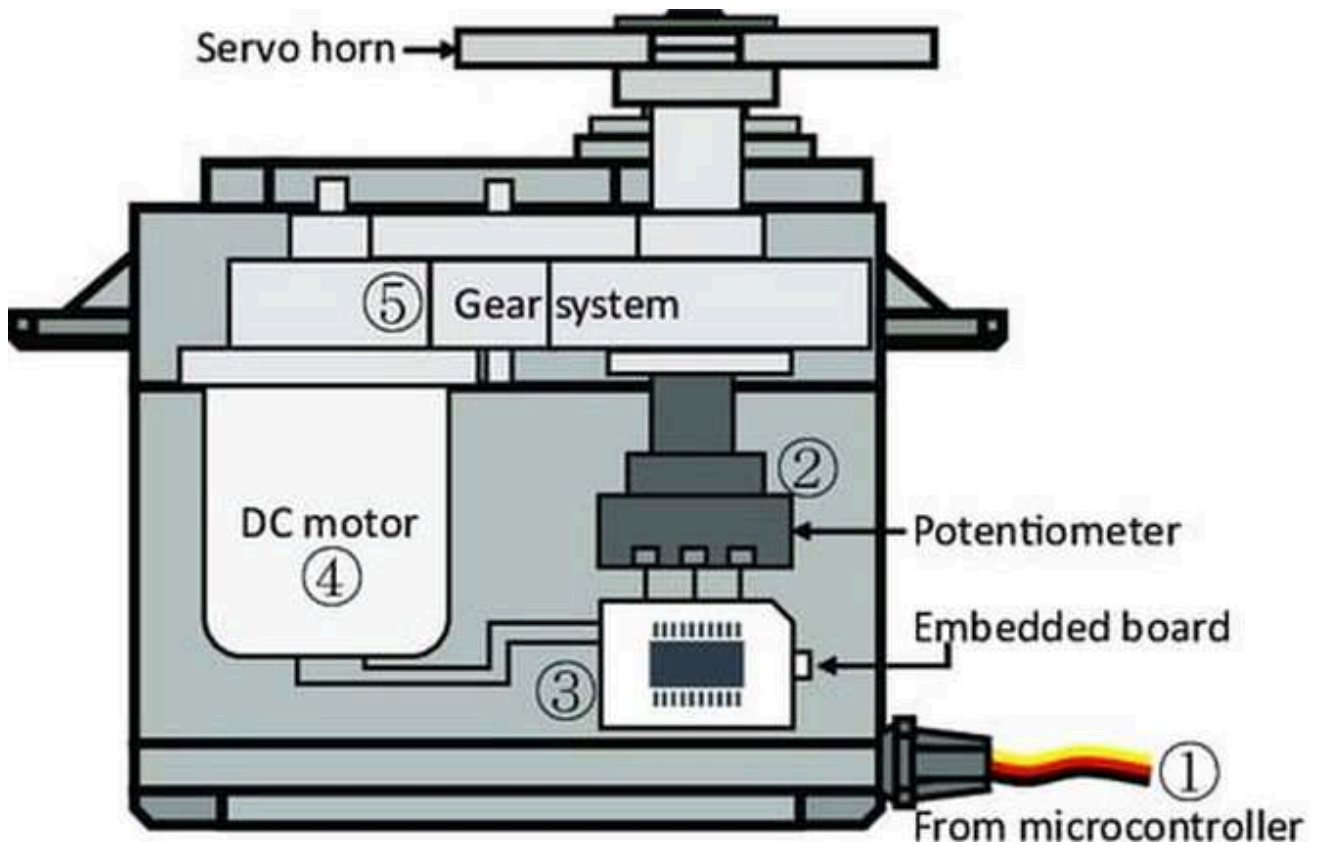
4.6.2 Servo

Descriere:

Servo, numit și **RC Servo Device**, este un motor cu feedback. De obicei, Servo efectuează un control precis al poziției și produce rezultate ridicate ridicare cuplu, care apare cel mai adesea în proiecte de robotică, mașini RC, avioane și avioane.



Structura internă:



- (1) Semnal(e): Primește semnalul de control de la microcontroler.
- (2) Potențiomtru: partea de feedback a servo. Măsoară poziția arborelui de ieșire.
- (3) Placă încorporată (controler intern): nucleul servo. El procesează semnalul de control extern și semnalul de feedback al poziției și acționează servo.
- (4) Motor de curent continuu: partea de execuție. Produce viteză, cuplu și poziție.
- (5) Sistem de angrenaje: Scalează ieșirile de la motor la ieșirea finală Unghiul în funcție de un anumit raport de transmisie.

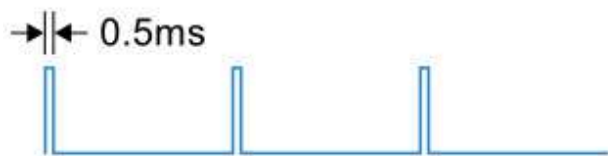
Conduceți servo:

Semnalul (semnalele) primește PWM pentru a controla ieșirea servo, iar poziția arborelui de ieșire se bazează direct pe ciclul de lucru al PWM.

De exemplu:

- Dacă trimitem un semnal cu o lățime a impulsului de 1,5 ms către Servo, arborele (claxon) se va roti în poziția de mijloc (90 °);
- Dacă lățimea impulsului = , arborele se rotește la minim (0°); 0.5ms
- Dacă lățimea impulsului = , arborele se rotește la maxim (180 °). 2.5ms

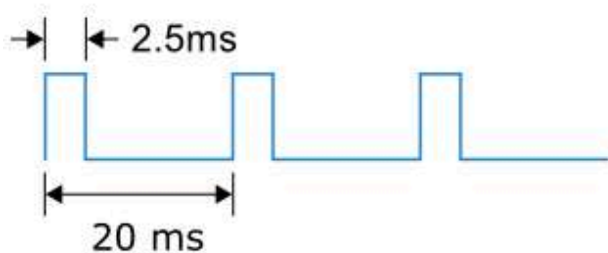
NOTĂ: Unghiul maxim variază în funcție de tipurile de servo. Unele sunt la 170° în timp ce altele sunt doar la 90°. În ciuda acestui fapt, servo-urile vor mișca de obicei un jumătate (din maxim) dacă primesc un semnal cu lățimea impulsului de 1,5 ms.



0 degrees



90 degrees



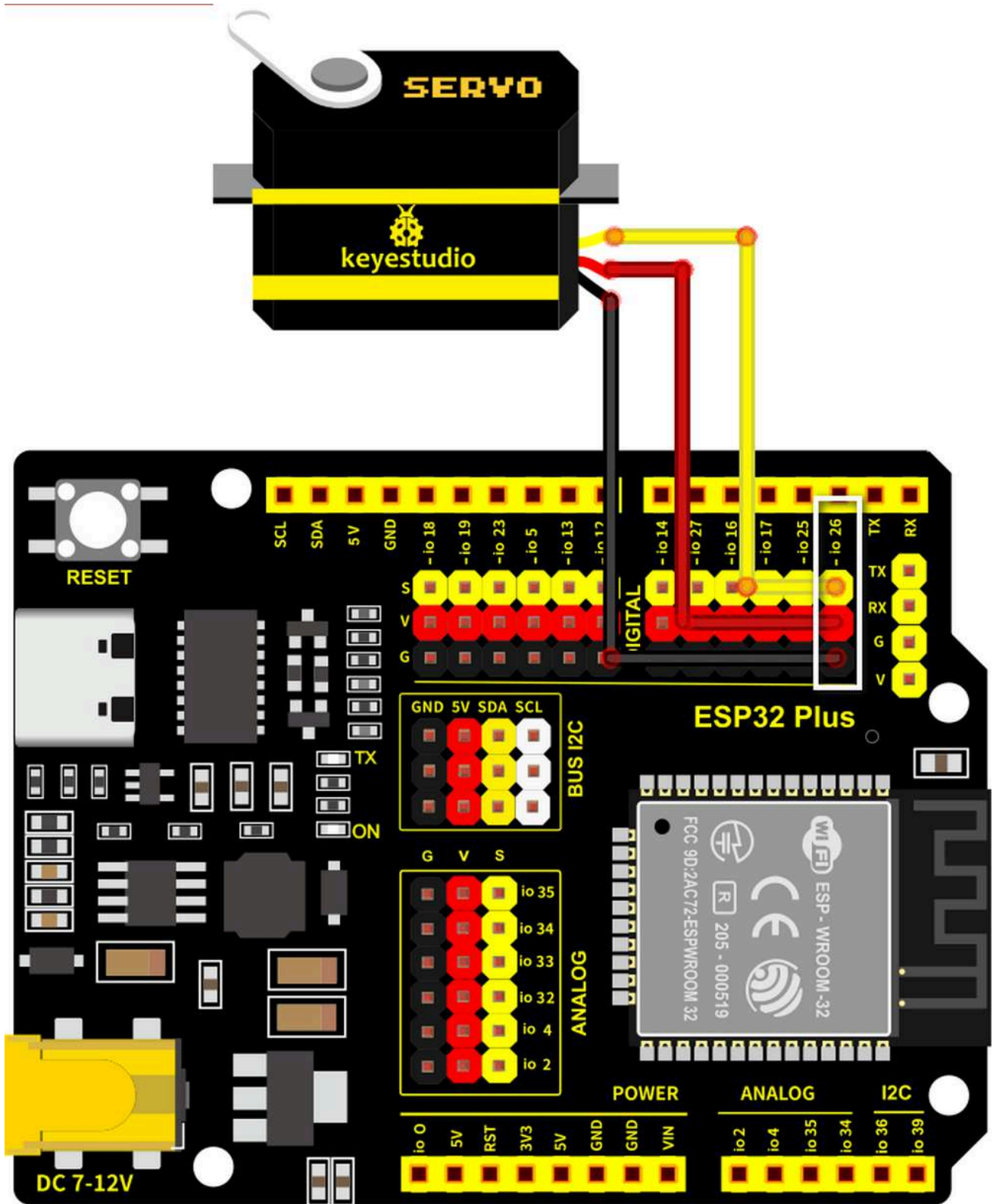
180 degrees

Perioada unui servo durează de obicei 20 ms și produce impulsuri la un frecvența . Majoritatea servomotoarelor funcționează normal la 40 ~ 200 Hz. 50Hz

Schema:

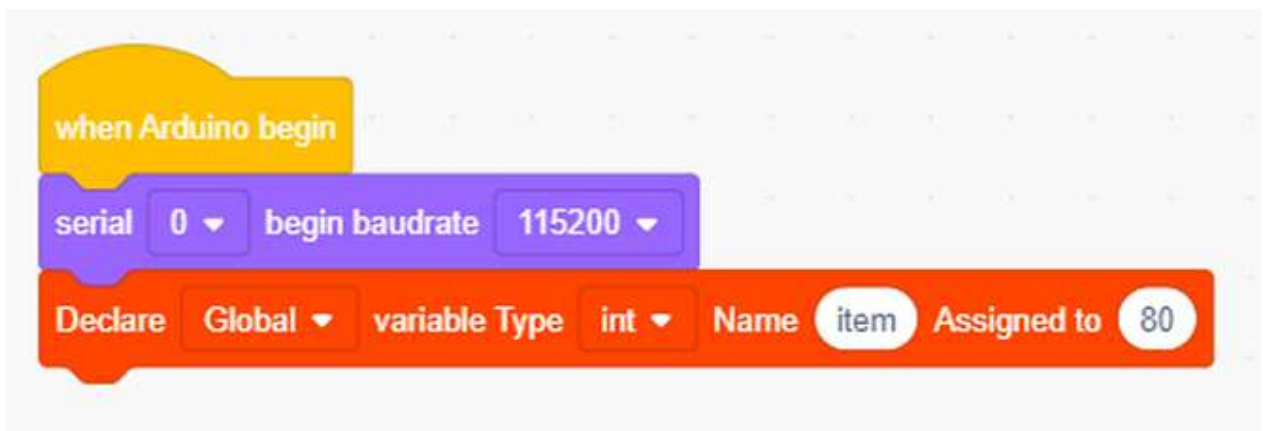
Conectați servo la io26.

Atenție: Conectați galben la S (Semnal), roșu la V (Alimentare) și negru la GND. Nu le inversați!



Cod de testare:

- Inițializați portul serial și definiți un element variabil cu un atribuire de 80.



when Arduino begin

serial 0 begin baudrate 115200

Declare Global variable Type int Name item Assigned to 80

- Setezi **elementul** la unghiul servo de la 80° la 180°, rotind 1° la fiecare 15ms.



forever

repeat 100

Change item variable mode ++

servo PIN# IO26 channel CH1 (LT0) degree variable item delay 15

- Servo se rotește cu 1° la fiecare 15ms, de la 180° la 80°.



repeat 100

Change item variable mode --

servo PIN# IO26 channel CH1 (LT0) degree variable item delay 15

Cod complet:

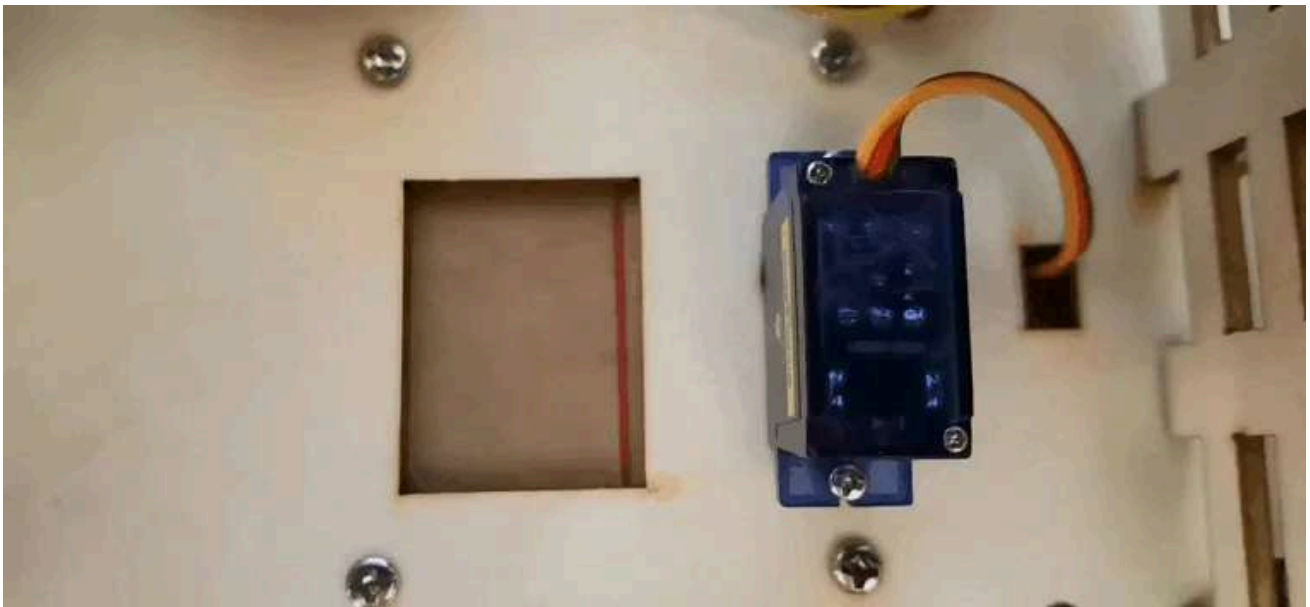


Rezultatul testului:

Cutia de alimentare este deschisă încet și apoi închisă, ceea ce este controlabil.

NOTĂ: Servo SG90 se poate roti la 180°. Deoarece cutia de alimentare este mică, 100° de rotație este suficientă pentru a închide complet cutia.

- 80°: complet deschis
- 120°: pe jumătate deschis
- 180°: închide



ATENȚIE

****Nu puneți degetele în cutie pentru a evita ciugulirea! ****

Nu blocați ușa cu ceva pentru a evita deteriorarea servo!

4.6.3 Senzor cu ultrasunete

Descriere:



Specification:

Working Voltage: DC 3.3V-5V

Working Current: 50mA - 100mA

Working Frequency: 40KHz

Max Power: 0.5W Max Range: 3m

Min Range: less than 4cm

Measuring Angle: less than 15 degree

Trigger Input Signal: 10 μ S TTL pulse

Interface: 4 pin port with 2.54mm space

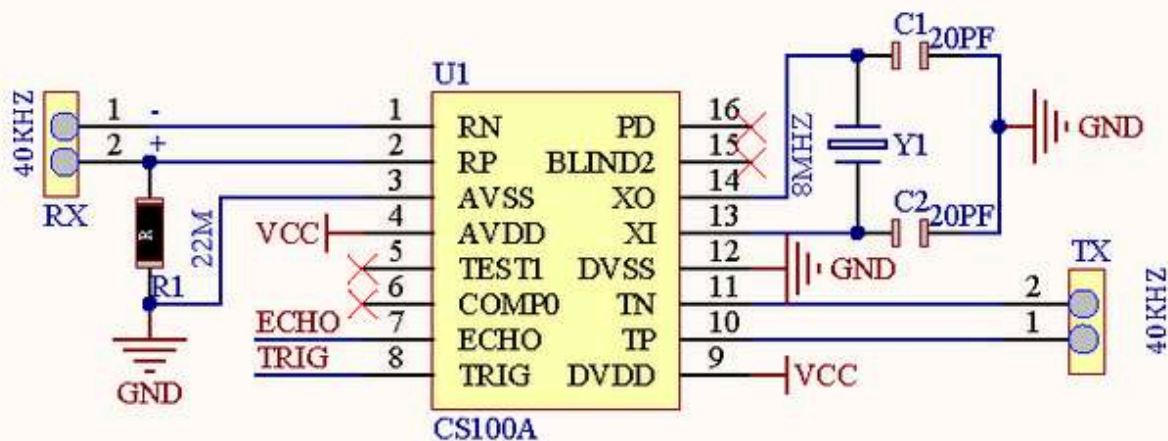
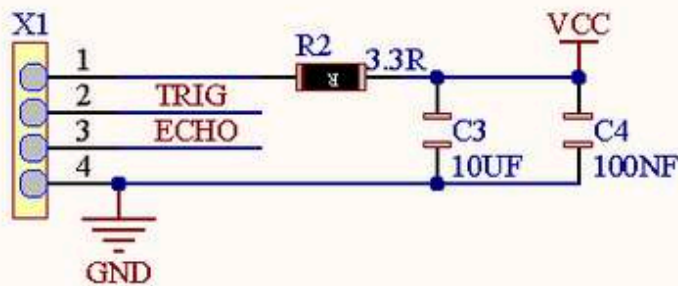
Position Holes: the diameter is 3mm

Working Temperature: -10 $^{\circ}$ C - +60 $^{\circ}$ C

Size: 49mm*22mm*19mm

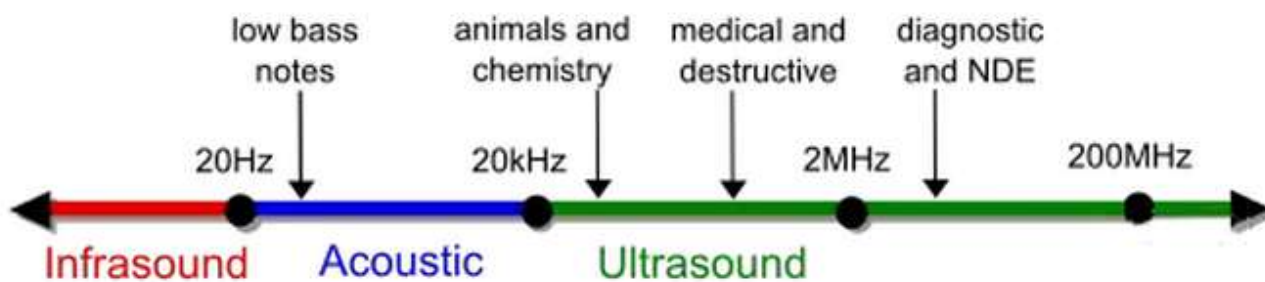
Environmental attributes: ROHS

Diagramă schematică:



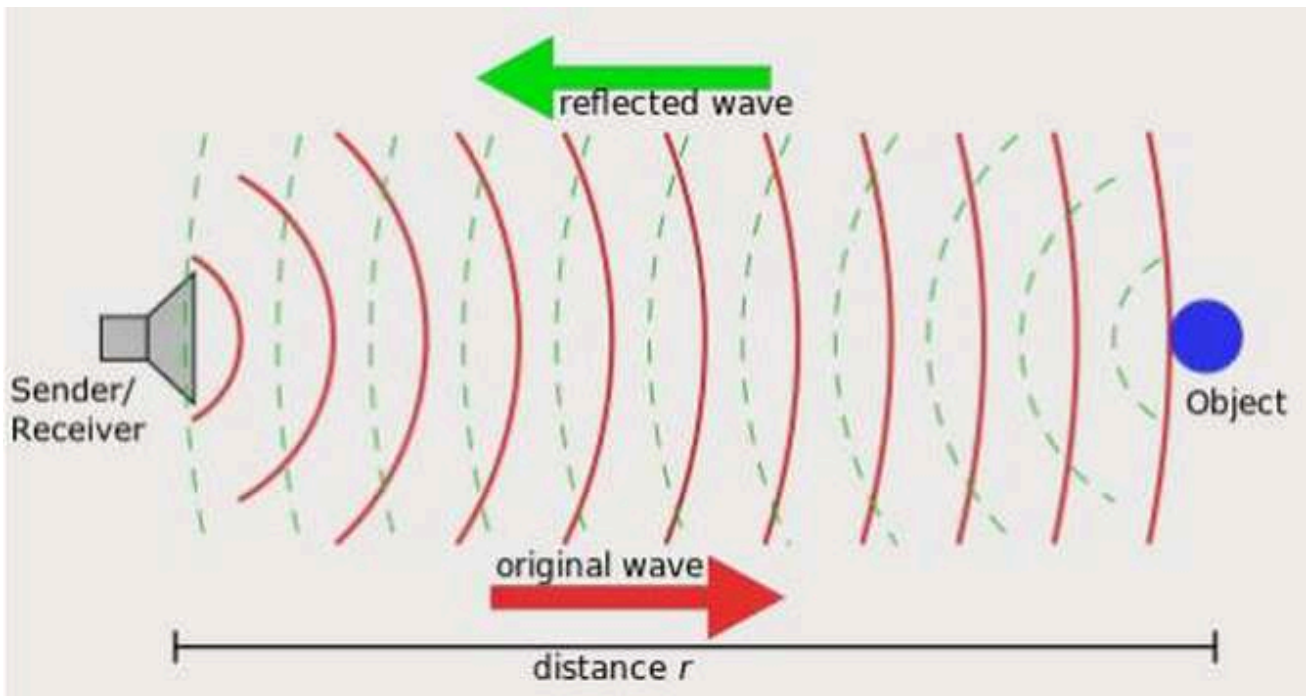
Frecvența undelor sonore pe care omul le poate auzi este de 20Hz ~ 20KHz, în timp ce undele ultrasonice sunt dincolo de acest interval.

Ultrasunete:



Modulul cu ultrasunete convertește electricitatea și undele ultrasonice în fiecare altele prin efect piezoelectric și, de asemenea, transmite și primește undă ultrasonică.

Acest tip de undă are directivitate, penetrare puternică și concentrarea energiei sonore.



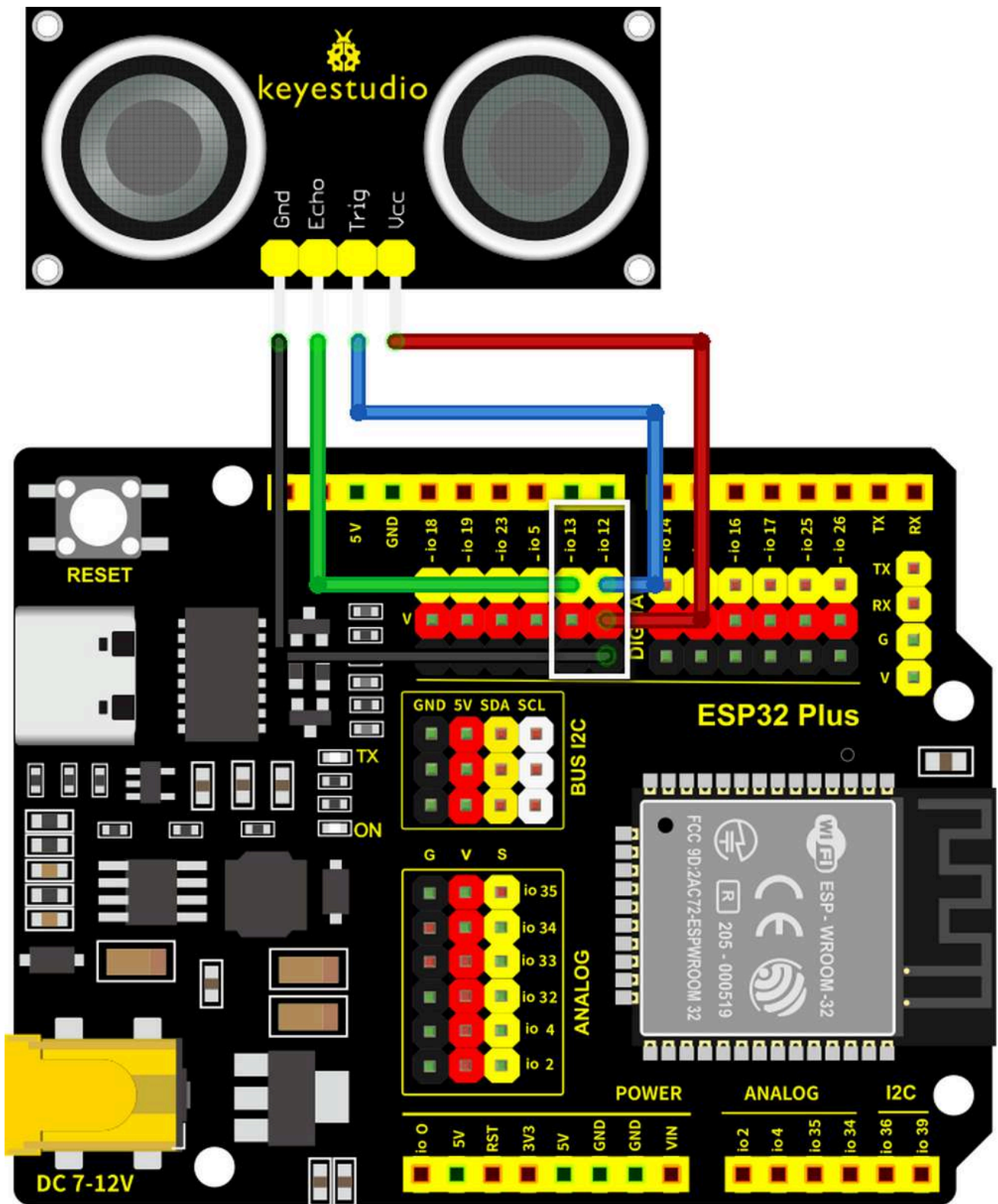
În acest sistem de simulare cu ultrasunete, programăm mai întâi pe MCU (ESP32 de dezvoltare) pentru a genera o undă pătrată originală la 40KHz și Conduceți modulul cu ultrasunete pentru a-l emite. Imediat, modulul calculează distanța până la obiect după primirea undă (Echo) amplificată și modelată de circuit. În acest document, se înregistrează durata emisiilor și a reflexiei și calculează distanța în funcție de diferența de fus orar.

Pur și simplu, MCU controlează modulul pentru a emite unde ultrasonice care sunt ricoșate înapoi după ce a întâlnit obstacole și este primit de modul. Cel diferența de timp dintre ele este un factor important în calcularea distanța (viteza de propagare a sunetului în aer este de 340 m/s).

Schema:

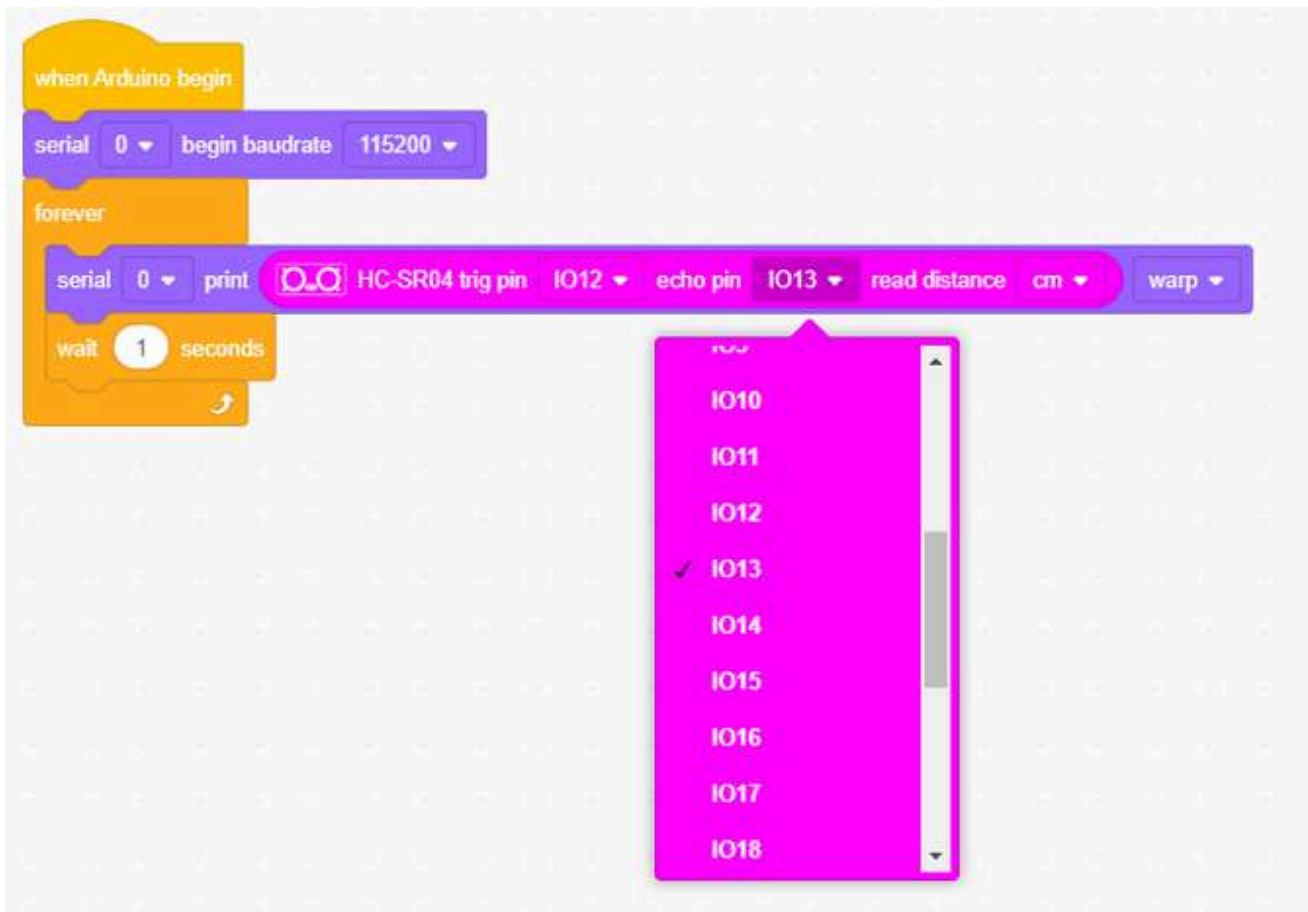
Conectați modulul Echo of Ultrasonic la io13 și Trig la io12.

Atenție: Conectați galben la S (Semnal) și roșu la V (Putere). Nu inversați-le!



Cod de testare:

Setați pinul corect: Trig la pin io12; Echo pentru a fixa io13.



Rezultatul testului:

În acest kit, intervalul de detectare este de 3 ~ 8 cm.

Deschideți monitorul serial și observați.



4.6.4 Sistem inteligent de hrănire

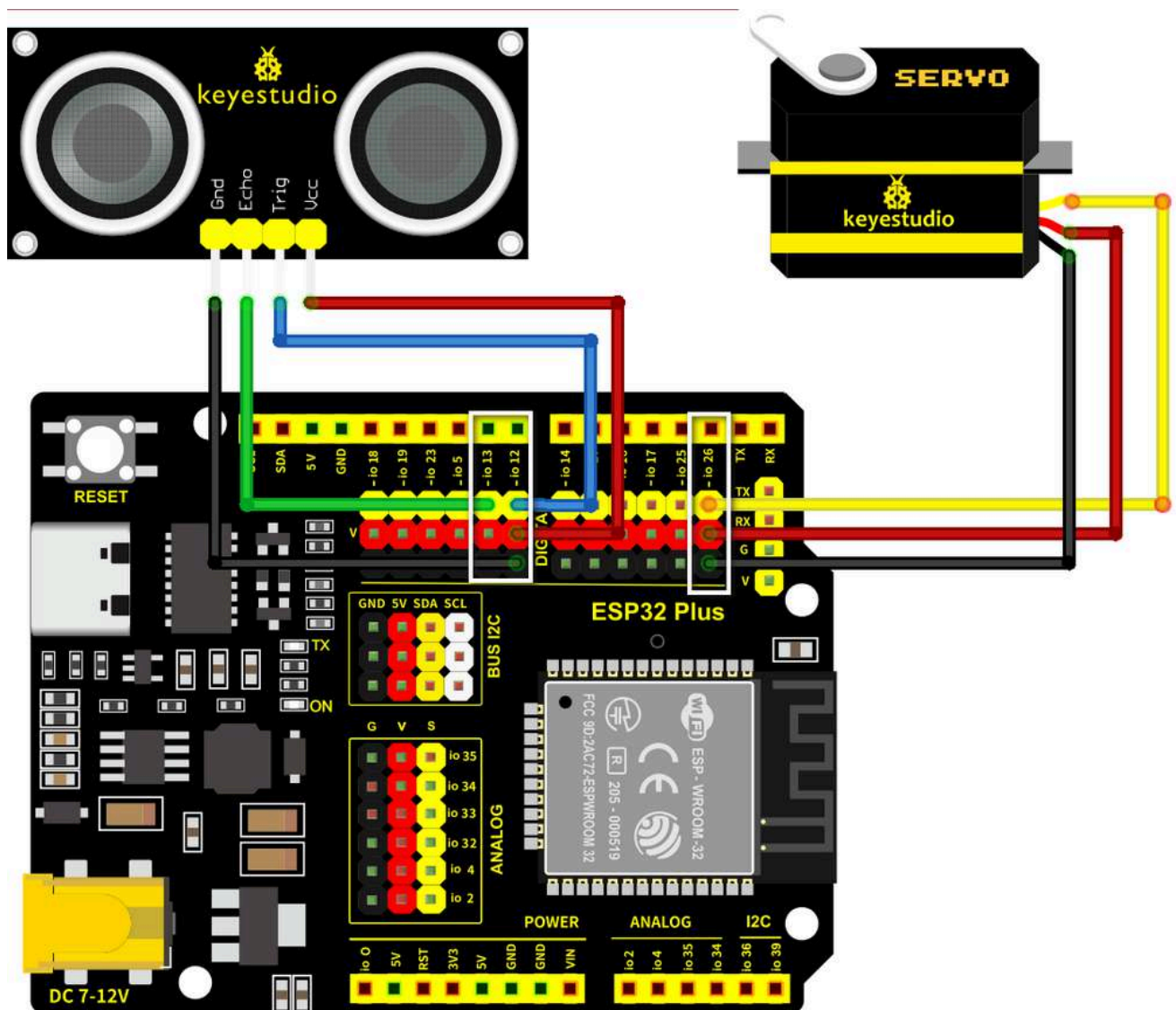
Descriere:

Sistemul inteligent de hrănire hrănește în mod inteligent păsările domestice printr-un modul cu ultrasunete și un servo. Primul detectează distanța până la animale, în timp ce ultima comandă să deschidă sau să închidă cutia de hrănire. Când Un animal de companie este detectat aproape de cutie, servo îl deschide pentru a se hrăni.

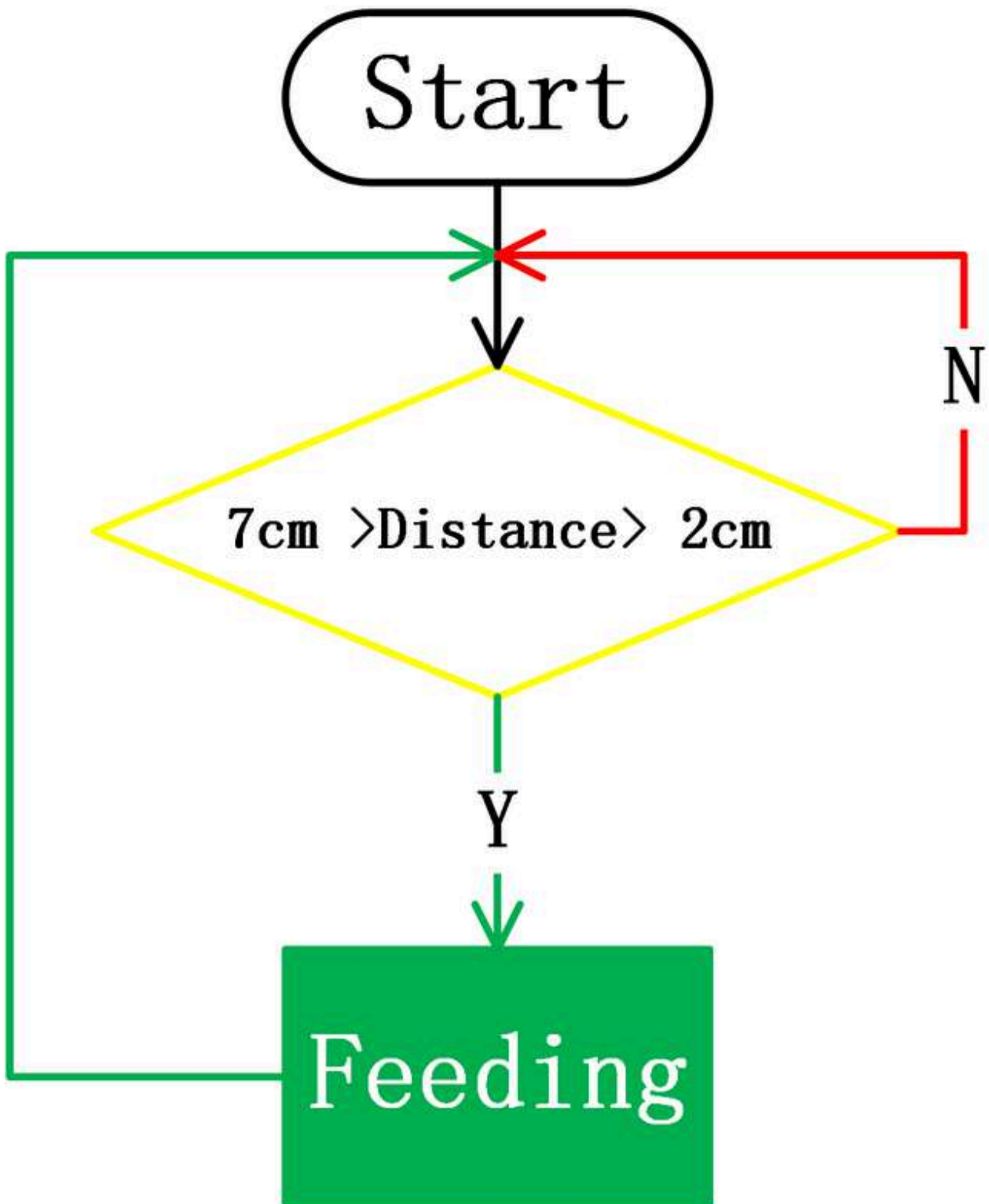
Schema:

Conectați modulul Echo of Ultrasonic la io13 și Trig la io12; Conectați servoul la IO26.

Atenție: Conectați galben la S (Semnal), roșu la V (Alimentare) și negru la GND. Nu le inversați!



Cod de testare:



Cod:

- Inițializați portul serial. Definiți o variabilă și atribuiți-o la 180.

```
when Arduino begin
  serial 0 begin baudrate 115200
  Declare Global variable Type int Name item Assigned to 180
```

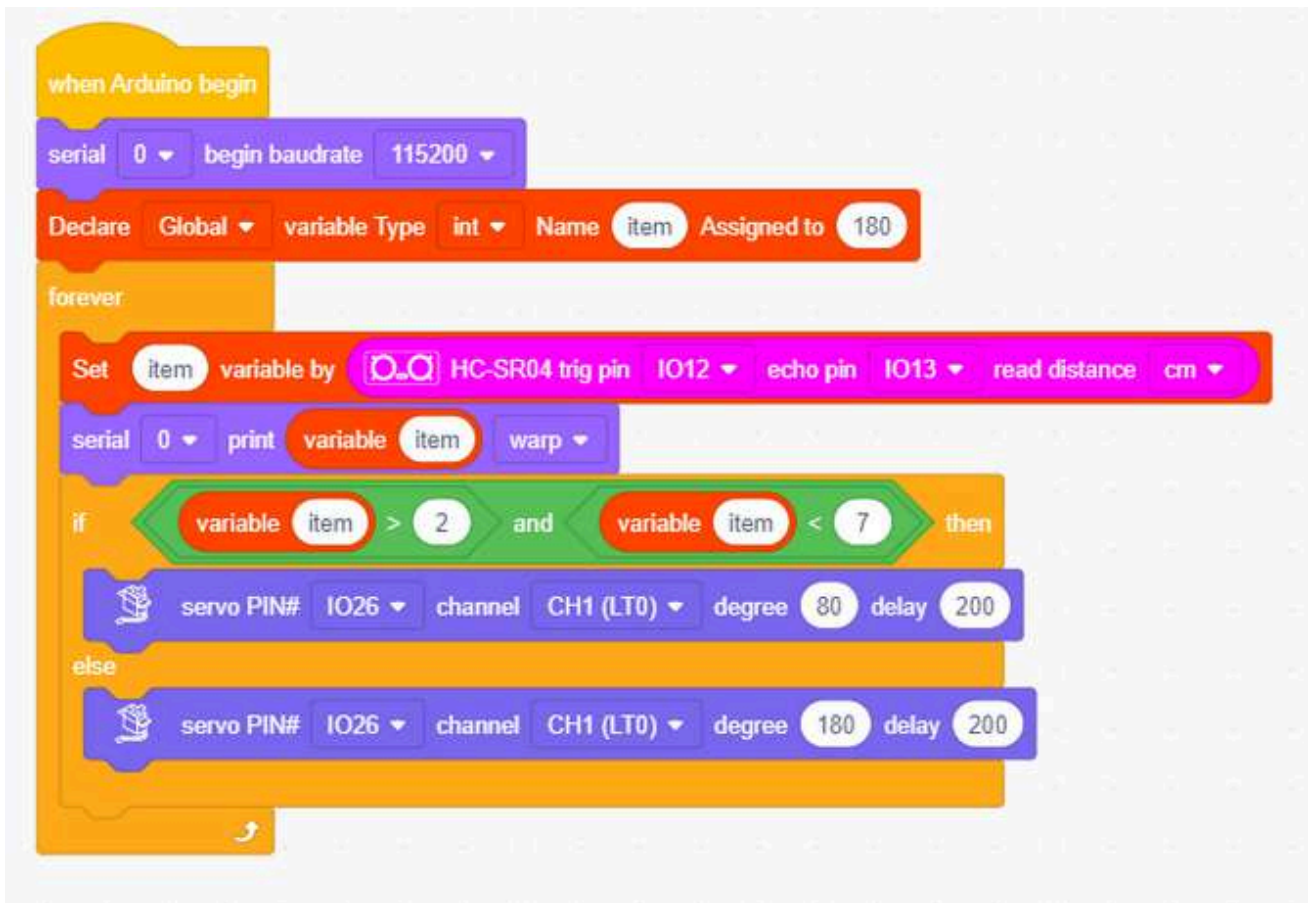
- Setează corect pinul și imprimăți valoarea primită.

```
forever
  Set item variable by HC-SR04 trig pin IO12 echo pin IO13 read distance cm
  serial 0 print variable item warp
```

- Determinați valoarea distanței detectate. Dacă este în limita a 2 cm ~ 7 cm, cutia de hrănire se va deschide.

```
if variable item > 2 and variable item < 7 then
  servo PIN# IO26 channel CH1 (LT0) degree 80 delay 200
else
  servo PIN# IO26 channel CH1 (LT0) degree 180 delay 200
```

Cod complet:



Rezultatul testului:

Când este detectat un animal, deschideți cutia de hrănire.

ATENȚIE

****Nu puneți degetele în cutie pentru a evita ciugulirea! ****

Nu blocați ușa cu ceva pentru a evita deteriorarea servo!

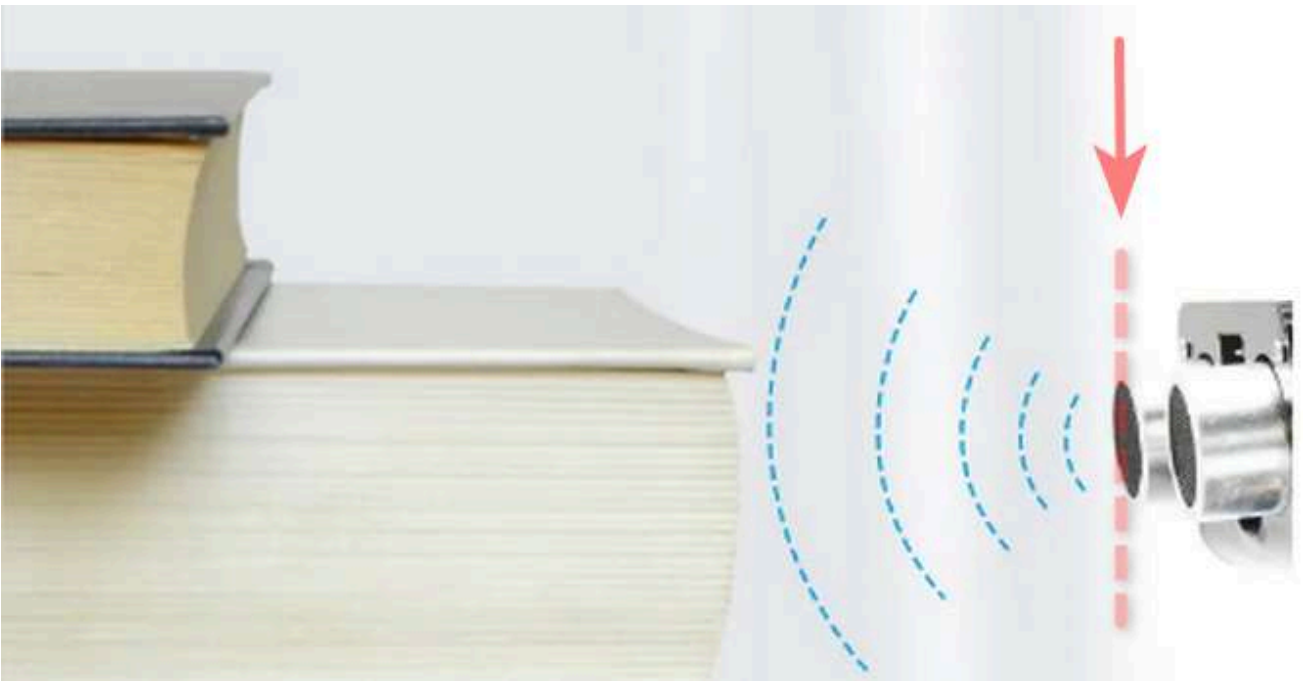
4.6.5 ÎNTREBĂRI FRECVENTE

Î: Servo nu funcționează.

R: Poate fi blocat singur sau de fire atunci când montați placa inferioară. Înainte de instalare, vă rugăm să reglați mai întâi servoul la 180 °. Pentru cum, Vă rugăm să consultați instrucțiunile de instalare.

Î: Distanța detectată este inexactă.

R: Când detectați, vă rugăm să măsurați de la capul de transmisie. Aici Acest un detector de înaltă precizie, deci pot exista erori.



4.7 Proiect: Sistem de control al temperaturii

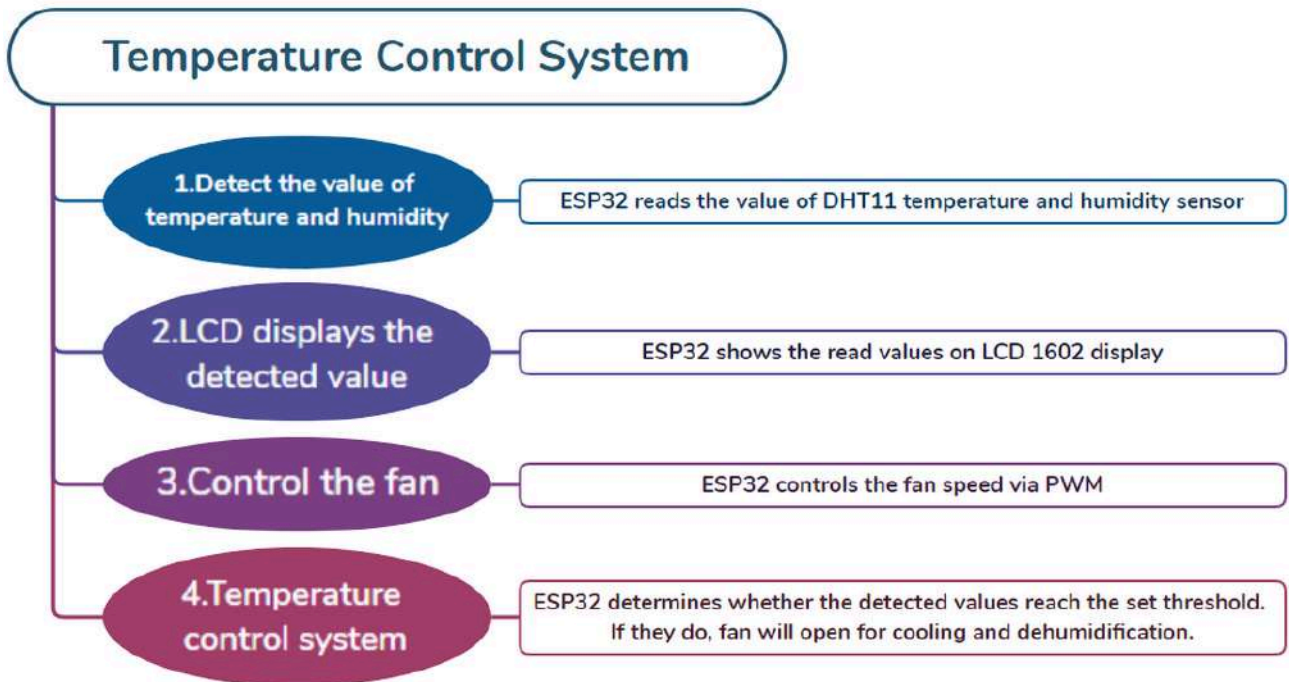
În acest proiect, vom demonstra cum să folosim temperatura și umiditatea senzor, ventilator și afișaj LCD1602 pentru a constitui o temperatură inteligentă și sistem de control al umidității.

Sistemul măsoară temperatura și umiditatea ambiantă și controlează ventilatorul pentru a se răcește după cum este necesar. Când temperatura depășește pragul setat, ventilatorul pornește automat pentru a reduce temperatura ambiantă sub setați valoare. Între timp, valorile actuale ale temperaturii și umidității vor să fie afișate pe LCD1602.

Prin urmare, realizează reglarea automată a temperaturii ambiante și umiditate, care este perfectă pentru proiectele care necesită aceste funcții.



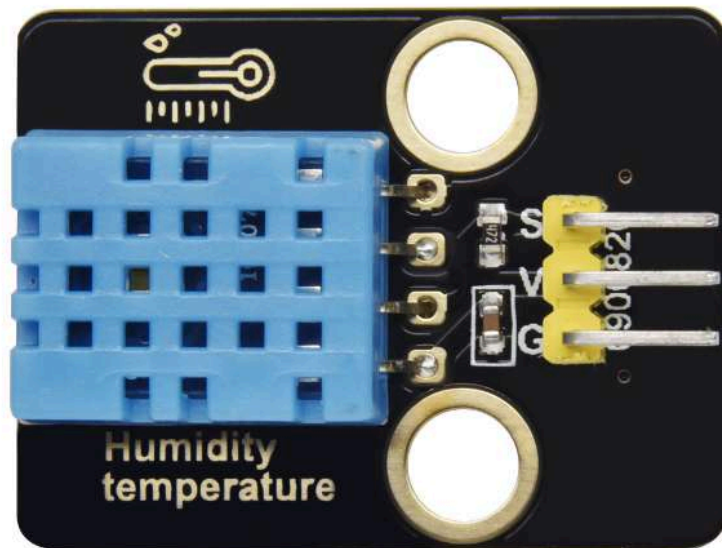
4.7.1 Diagrama de flux



4.7.2 Senzor de temperatură și umiditate

Descriere:

Senzorul de temperatură și umiditate DHT11 emite semnale digitale. El aplică și principiile de achiziție și conversie a semnalului analogic ca tehnologie de detectare a temperaturii și umidității, astfel încât să dispună de stabilitate pe termen lung și fiabilitate ridicată. În plus, senzorul integrează un senzor de umiditate rezistiv de înaltă precizie și un senzor de umiditate rezistiv senzor de temperatură termosensibil și este conectat cu un MCU de înaltă performanță.



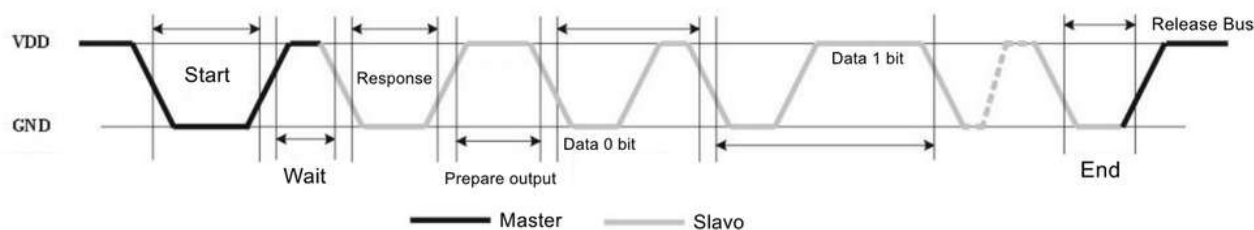
Comunicarea DHT11 înseamnă:

DHT11 comunică prin monobuz (o singură magistrală) care schimbă și controlează datele.

- Monobus transmite **bit de date**:
 - Formatul de date al monobusului: transmiteți date de 40 de biți de fiecare dată și high-bit mai întâi.
 - Umiditate de 8 biți valoare întreagă + umiditate de 8 biți valoare zecimală + 8 biți valoare întreagă de temperatură + valoare zecimală de temperatură de 8 biți + 8 biți paritate.
 - **NOTĂ: Valoarea zecimală a umidității este egală cu 0.**
- **Bit Paraty:**

- Umiditate de 8 biți valoare întreagă + umiditate de 8 biți valoare zecimală + 8 biți valoare întreagă de temperatură + valoare zecimală de temperatură de 8 biți.
- Paritatea de 8 biți este egală cu 8 biți de capăt ai rezultatului.

Diagrama de sincronizare:



****NOTĂ:****

Gazda citește întotdeauna valorile temperaturii și umidității pentru ultima măsurare de la DHT11. Prin urmare, dacă intervalul dintre două măsurătorile sunt lungi, vă rugăm să detectați consecutiv de două ori și să adoptați al doilea rezultat.

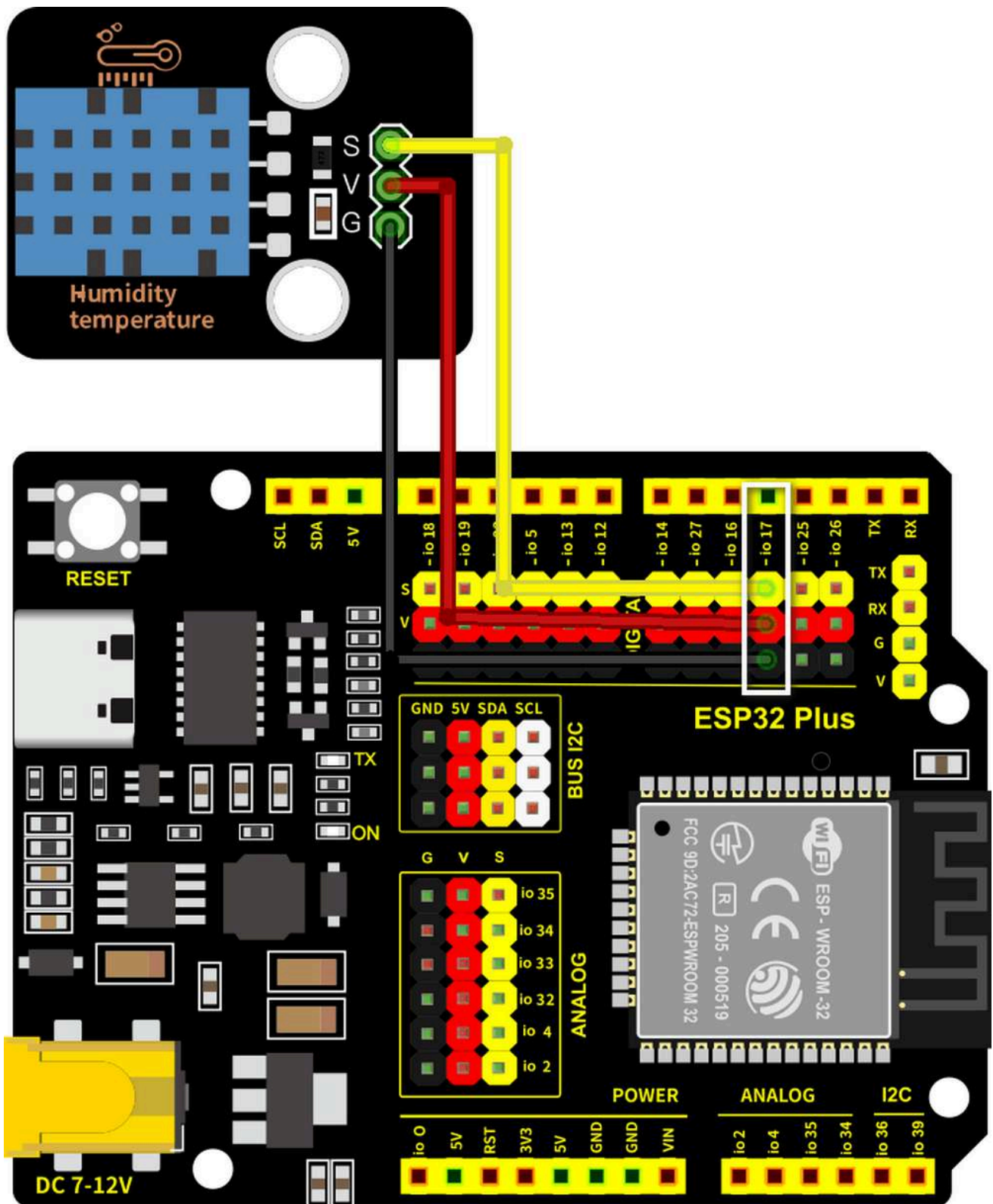
Pentru mai multe detalii, vă rugăm să vizitați site-ul oficial ASAIR:

<http://www.aosong.com/products-21.html>

Schema:

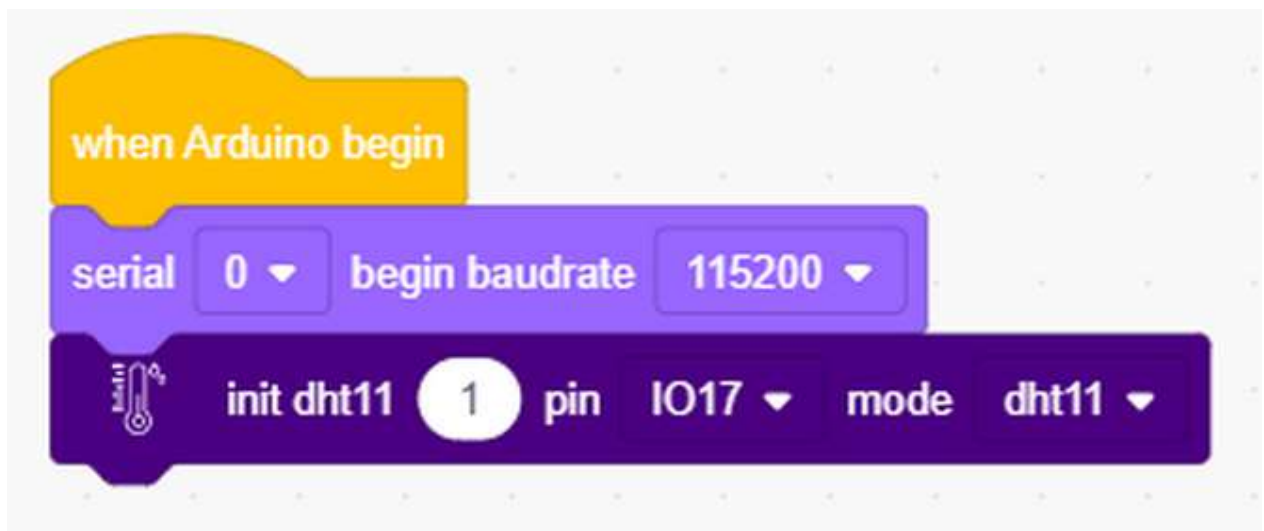
Conectați senzorul de temperatură și umiditate la io17.

Atenție: Conectați galben la S (Semnal), roșu la V (Alimentare) și negru la GND. Nu le inversați!



Cod de testare:

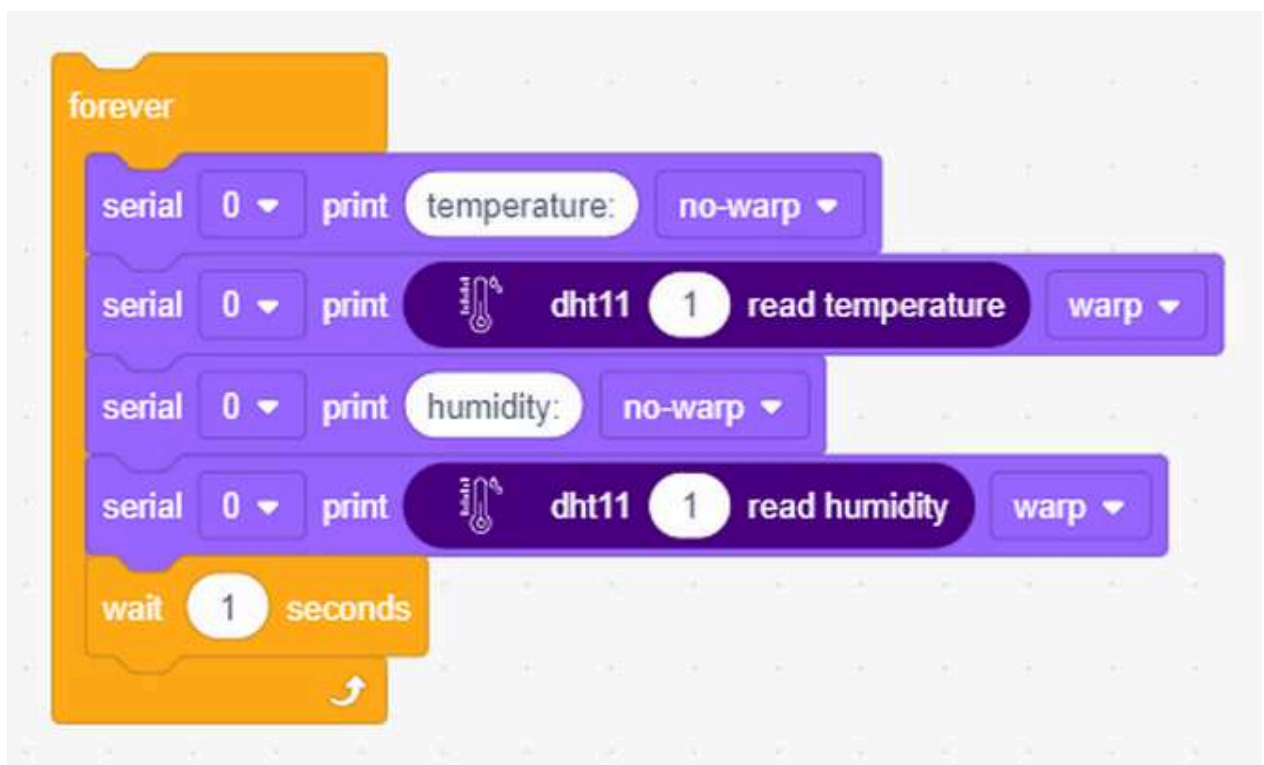
- Inițializați portul serial și senzorul.



when Arduino begin

- serial 0 begin baudrate 115200
- init dht11 1 pin IO17 mode dht11

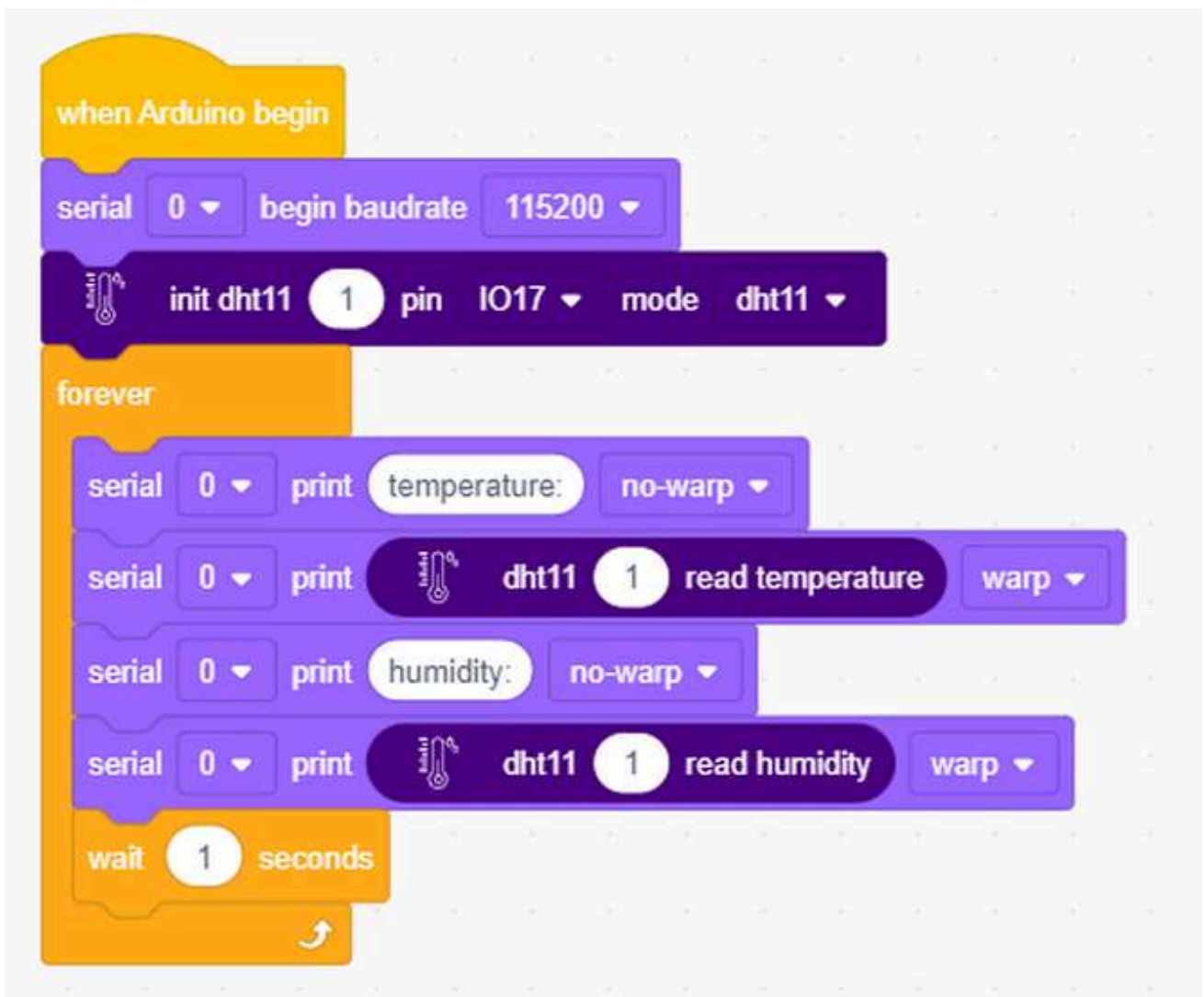
- Monitorul serial imprimă și reîmprospătează umiditatea și temperatura valori pe secundă.



forever

- serial 0 print temperature: no-warp
- serial 0 print dht11 1 read temperature warp
- serial 0 print humidity: no-warp
- serial 0 print dht11 1 read humidity warp
- wait 1 seconds

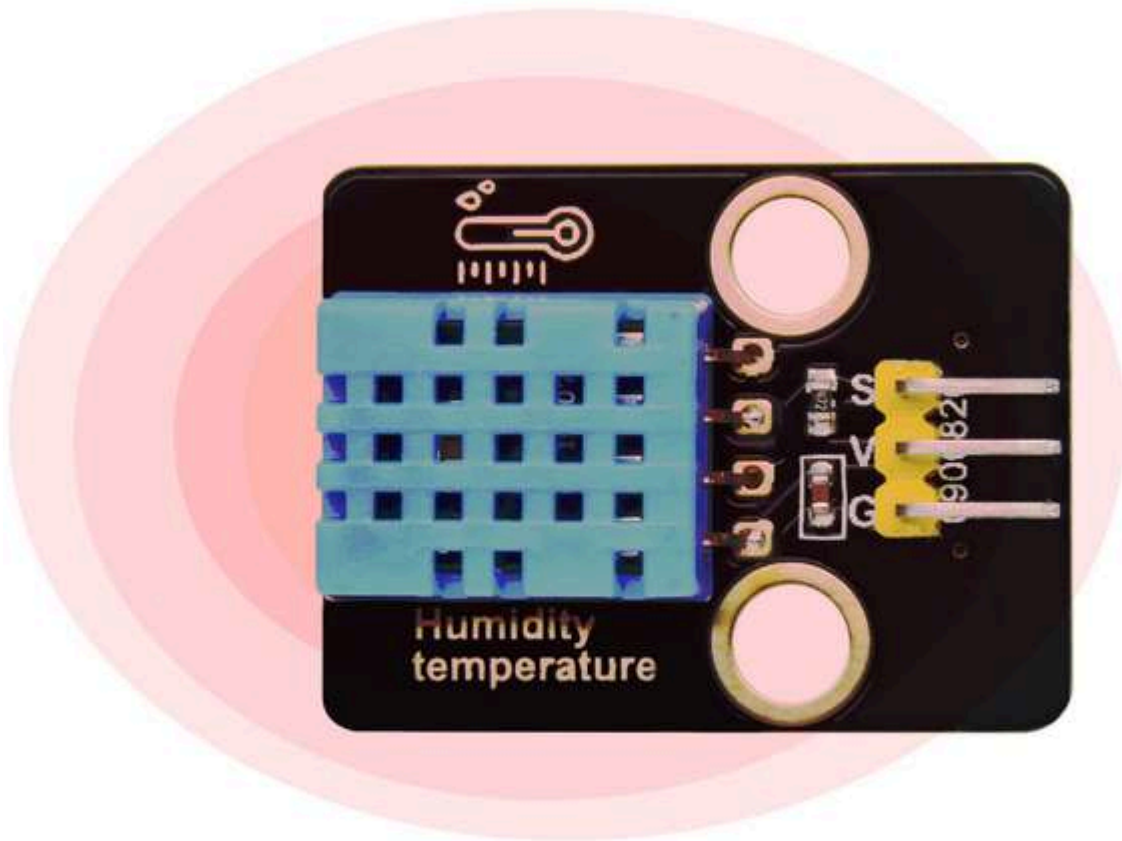
Cod complet:



```
when Arduino begin
  serial 0 begin baudrate 115200
  init dht11 1 pin IO17 mode dht11
  forever
    serial 0 print temperature: no-warp
    serial 0 print dht11 1 read temperature warp
    serial 0 print humidity: no-warp
    serial 0 print dht11 1 read humidity warp
    wait 1 seconds
```

The image shows a Scratch script for an Arduino project. It starts with a 'when Arduino begin' block, followed by 'serial 0 begin baudrate 115200'. Then, an 'init dht11 1 pin IO17 mode dht11' block initializes the sensor. A 'forever' loop contains four 'serial 0 print' blocks: 'temperature: no-warp', 'dht11 1 read temperature warp', 'humidity: no-warp', and 'dht11 1 read humidity warp'. The loop ends with a 'wait 1 seconds' block.

Rezultatul testului:



Detection Area

Deschideți monitorul serial și veți vedea temperatura curentă și valoarea umidității.

```
temperature:29.70
humidity:71.10
temperature:29.70
humidity:71.10
temperature:29.60
humidity:70.60
temperature:29.60
humidity:70.60
temperature:29.60
humidity:70.10
temperature:29.60
humidity:70.10
temperature:29.60
humidity:69.80
```

Buadrate 115200 ▾

End of line LF & CR ▾

Hex form

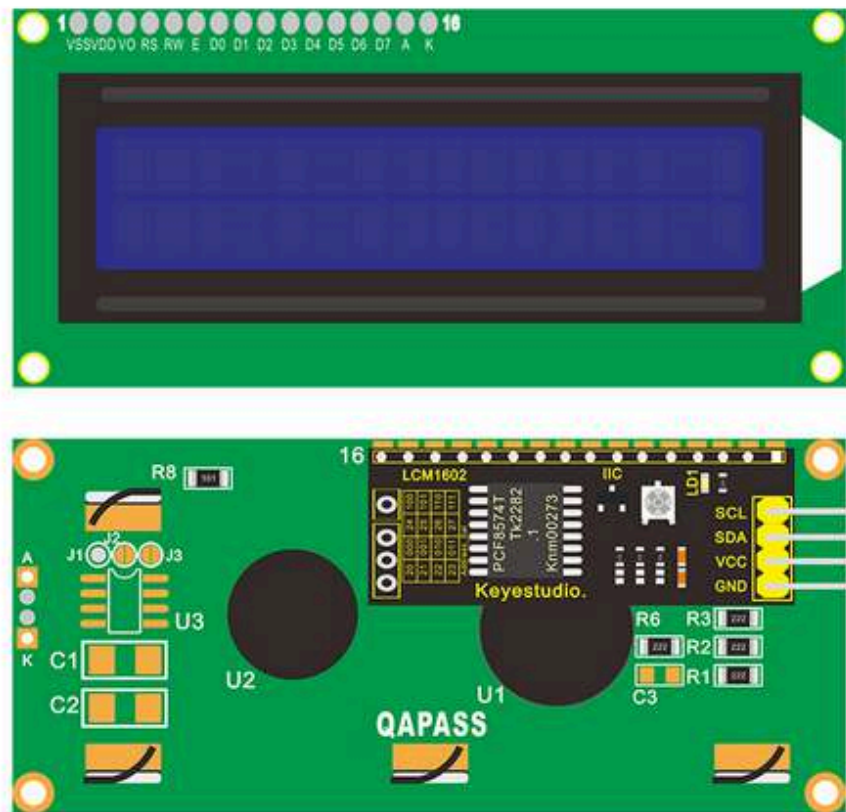
Auto scroll

Ultima ▾

4.7.3 Modul LCD 1602

Descriere:

LCD 1602 posedă un standard cu 14 pini (fără iluminare de fundal) sau cu 16 pini (cu iluminare din spate), salvând pinii MCU. Afişajul său conduce IC pentru a realiza controlul I2C.



Specifications

Power supply range: 5V

Working current: <130mA

Recommended ambient temperature: -10°C ~ 50°C

Product size: 80mm * 36mm

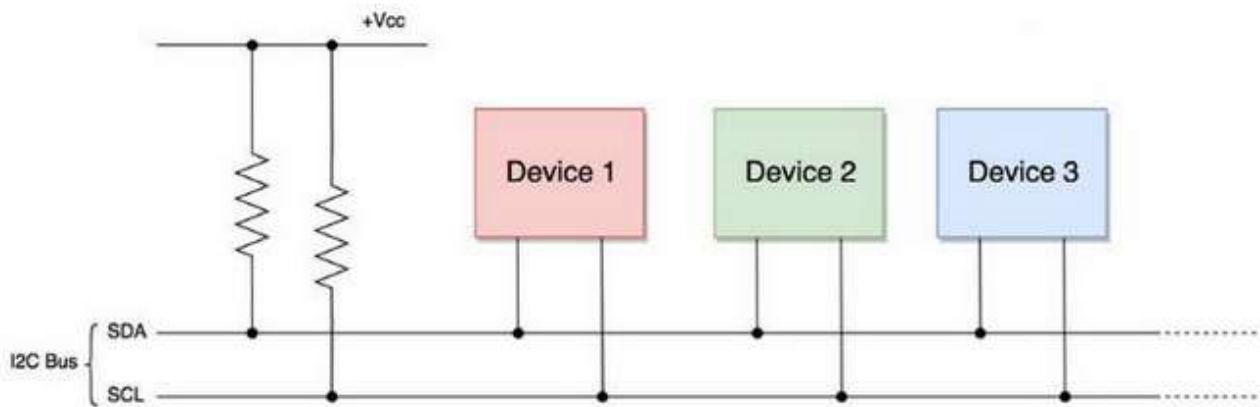
IIC address: 0x27

Comunicare serială I2C:

I2C, cunoscută pe deplin sub denumirea de circuit interintegrat (IIC) sau Interfața cu două fire (TWI) este o magistrală duală utilizată în mod obișnuit (o gazdă și un slave), care este dezvoltat de Phillips Semiconductor (achiziționat de US NXP Semiconductor

Cel mai mare avantaj este că doar două fire completează transmisia de date, ceea ce simplifică în mare măsură circuitele. În total, autobuzul I2C poate conectați 127 de noduri în paralel, astfel încât să accepte mai multe gazde și sclavi.

În general, sursa de alimentare externă este inutilă pentru sclavi, deoarece magistrala I2C transmite-le puterea:



Magistrala I2C transmite date prin transmisie de date pe 8 biți. De obicei Datele de un octet sunt compuse din nouă semnale de ceas, dintre care opt transmit și ultimul marchează sfârșitul transmiterii.

Mai mult, magistrala I2C acceptă transmisia de date pe mai mulți octeți prin repetarea proces de mai sus continuu.

Protocolul I2C constă practic din:

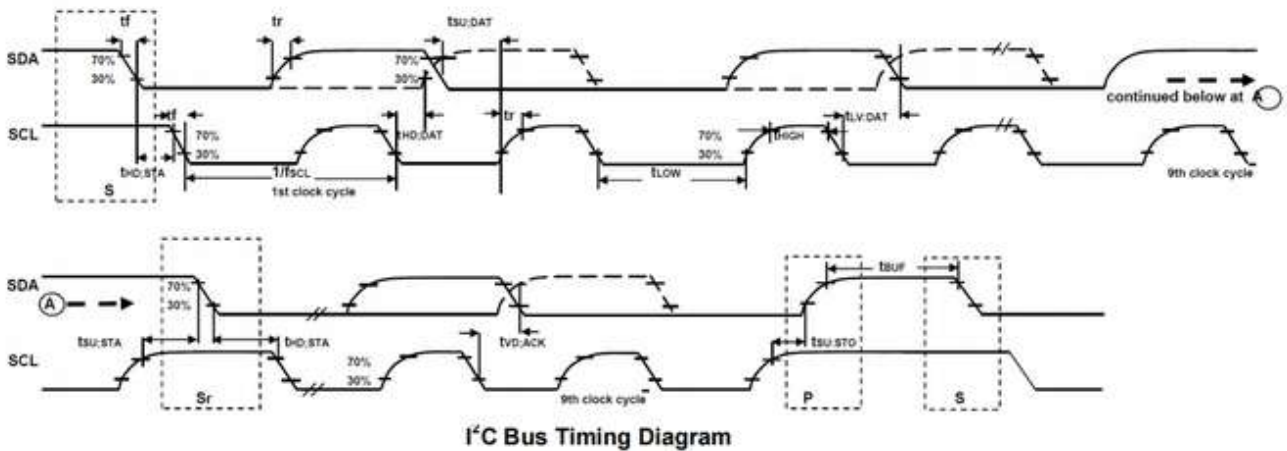
- **Semnal de pornire:** Înainte de transmisie, emițătorul transmite o pornire semnal pentru a informa receptorul despre punctul de plecare.
- **Adresă:** Notifică destinatarul căruia îi sunt trimise datele.
- **Date:** Se transmite câte un octet de fiecare dată și bit cu bit.
- **Semnal de sfârșit:** La terminarea transmisiei, expeditorul încheie datele cu un semnal de sfârșit pentru a informa receptorul că procesul s-a încheiat.

Diagrama de sincronizare a protocolului serial:

Pentru mai multe detalii, vă rugăm să vizitați site-ul oficial: <https://www.nxp.com/>

Parameters	Conditions	Min	Typical	Max	Units	Notes
I²C TIMING						
f_{SCL} , SCL Clock Frequency	I ² C FAST-MODE			400	kHz	
$t_{HD,STA}$, (Repeated) START Condition Hold Time		0.6			μ s	
t_{LOW} , SCL Low Period		1.3			μ s	
t_{HIGH} , SCL High Period		0.6			μ s	
$t_{SU,STA}$, Repeated START Condition Setup Time		0.6			μ s	
$t_{HD,DAT}$, SDA Data Hold Time		0			μ s	
$t_{SU,DAT}$, SDA Data Setup Time		100			ns	
t_r , SDA and SCL Rise Time	C_b bus cap. from 10 to 400pF	$20+0.1C_b$		300	ns	
t_f , SDA and SCL Fall Time	C_b bus cap. from 10 to 400pF	$20+0.1C_b$		300	ns	
$t_{SU,STO}$, STOP Condition Setup Time		0.6			μ s	
t_{BUF} , Bus Free Time Between STOP and START Condition		1.3			μ s	
C_b , Capacitive Load for each Bus Line			< 400		pF	
$t_{VD,DAT}$, Data Valid Time				0.9	μ s	
$t_{VD,ACK}$, Data Valid Acknowledge Time				0.9	μ s	

Note: Timing Characteristics apply to both Primary and Auxiliary I²C Bus



Vă punem la dispoziție un fișier de bibliotecă **Wire.h** pe Arduino pentru I2C protocol, în care funcțiile pot fi apelate direct pentru a comunica cu Dispozitive I2C/TWI.

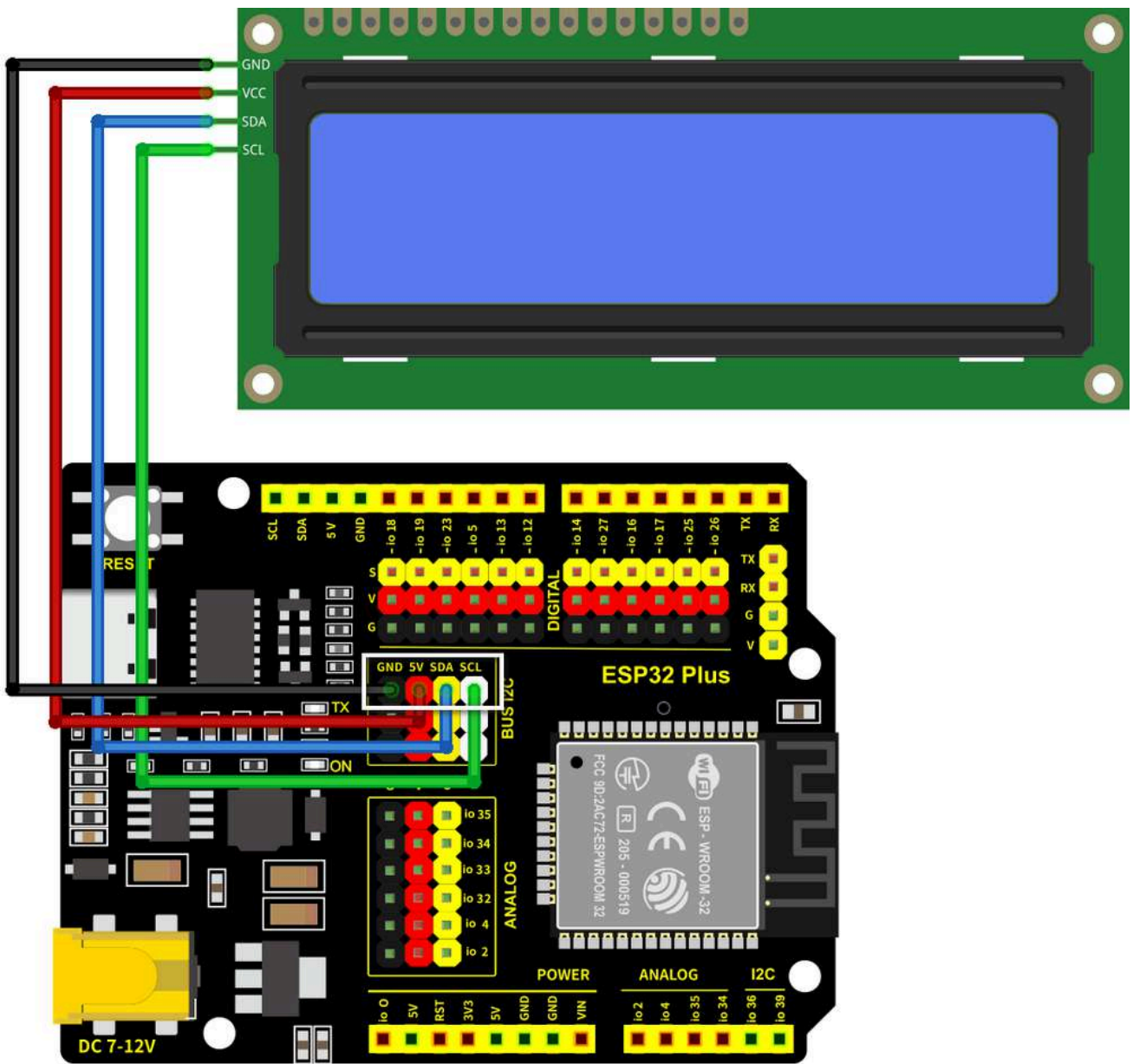
Pentru detalii despre bibliotecă, vă rugăm să consultați:

<https://www.arduino.cc/reference/en/language/functions/communication/wire/>

Schema:

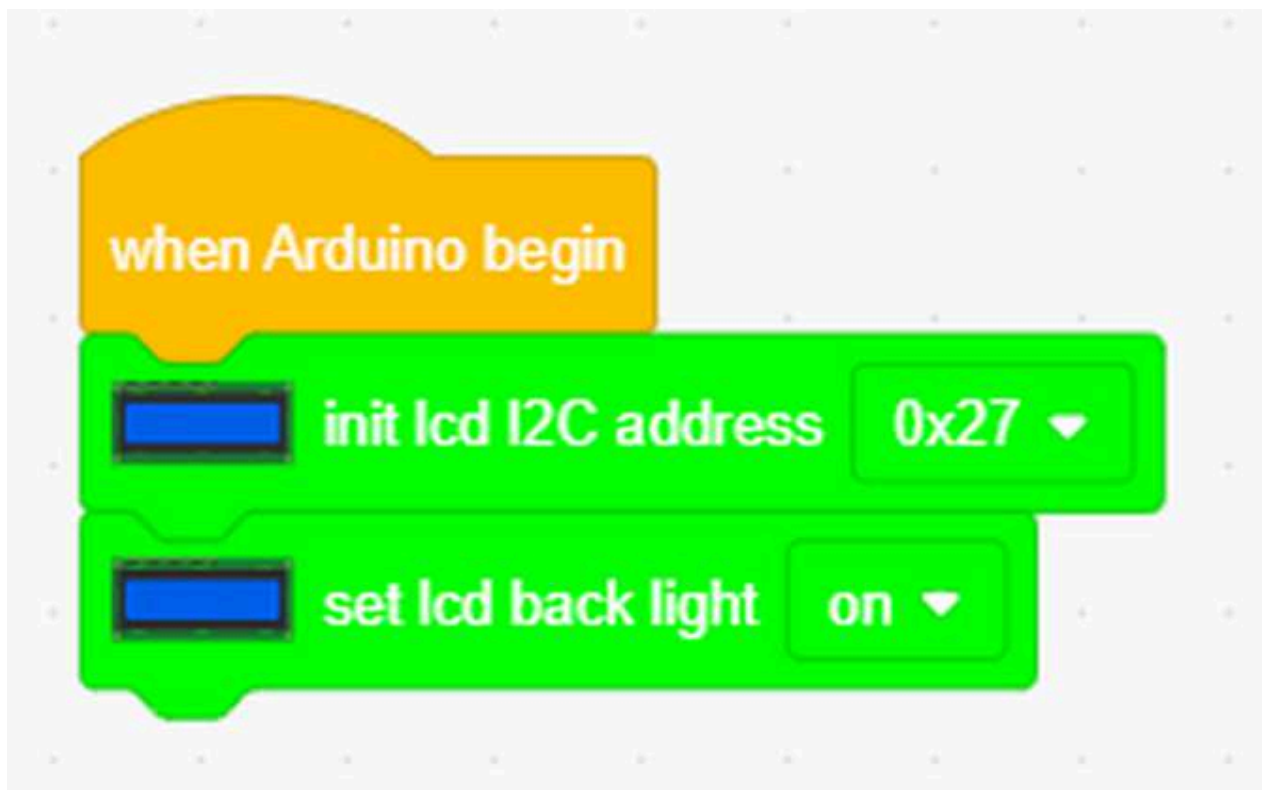
Conectați ecranul LCD la I2C BUS așa cum se arată mai jos.

Atenție: Conectați galben la S (Semnal), roșu la V (Alimentare) și negru la GND. Nu le inversați!



Cod de testare:

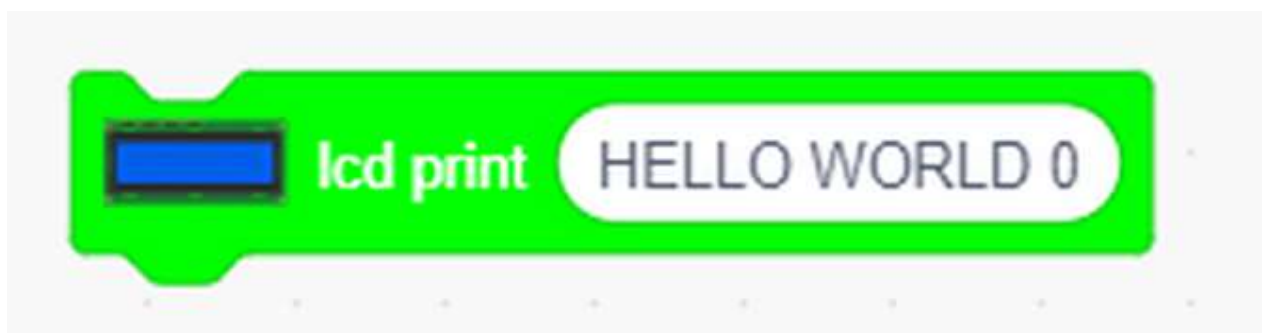
- Inițializați adresa I2C a LCD-ului și porniți lumina de fundal.



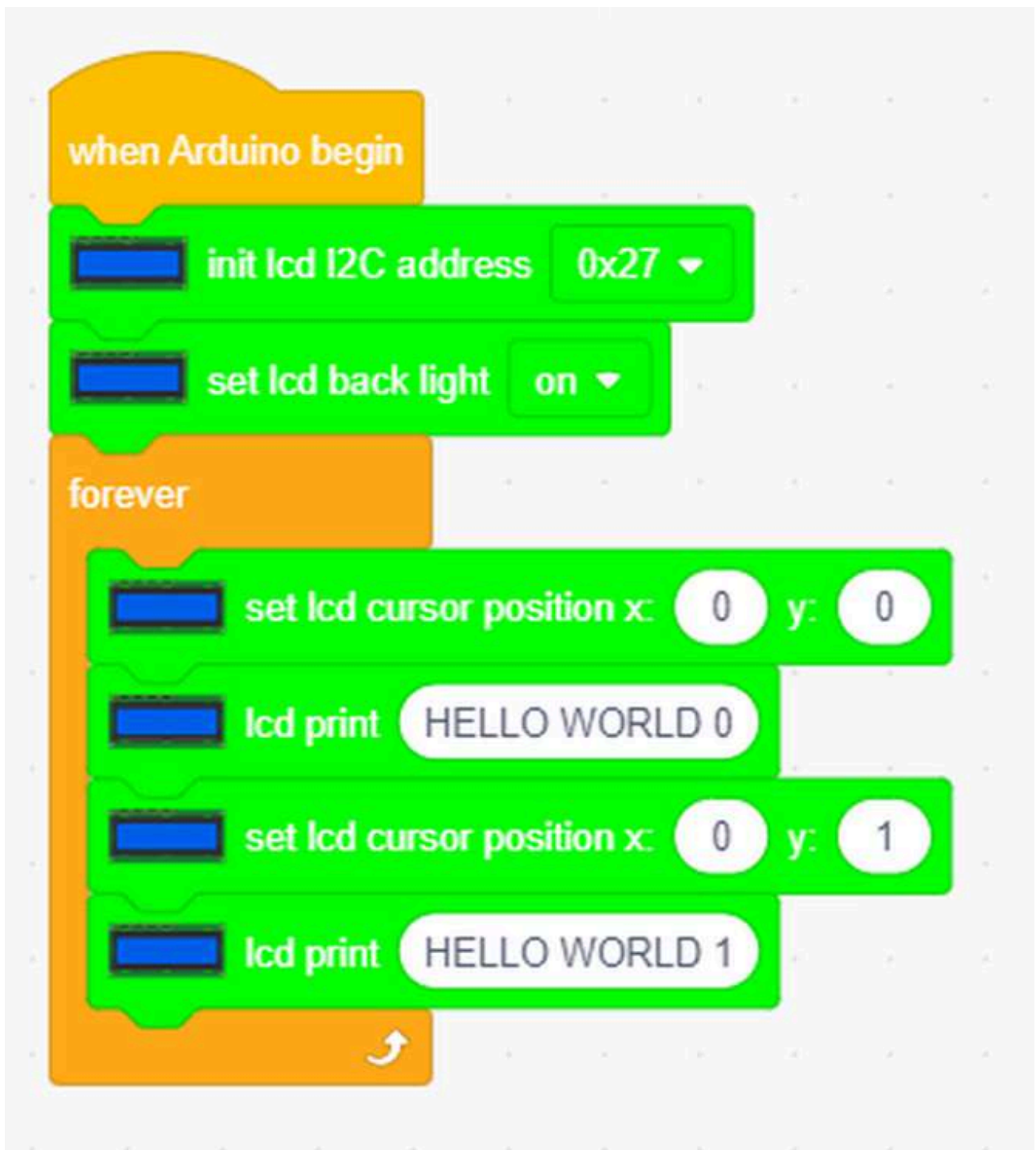
- Setezi poziția cursorului LCD pe axele X și Y (axa X afișează un maxim 16 caractere, iar axa Y afișează maximum 2 coloane).



- Introduceți conținutul imprimat (nu mai mult de 16 caractere, altfel nu va fi complet).

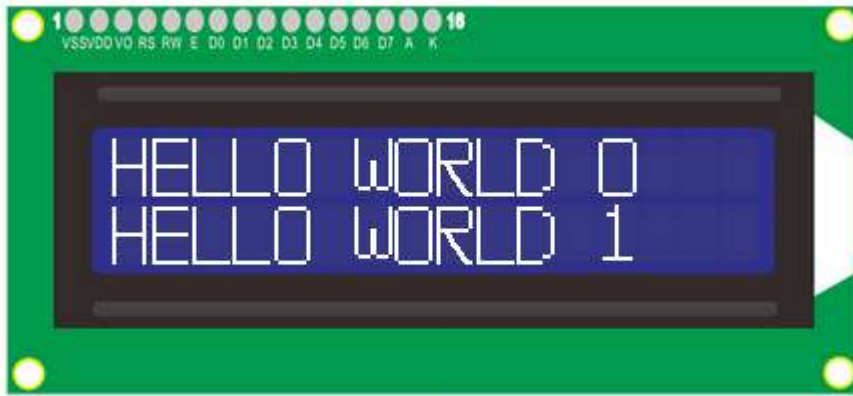


Cod complet:



Rezultatul testului:

LCD1602 deschide lumina de fundal și afișează "HHELLO WORLD 0" și "BUNĂ LUMEA 1".

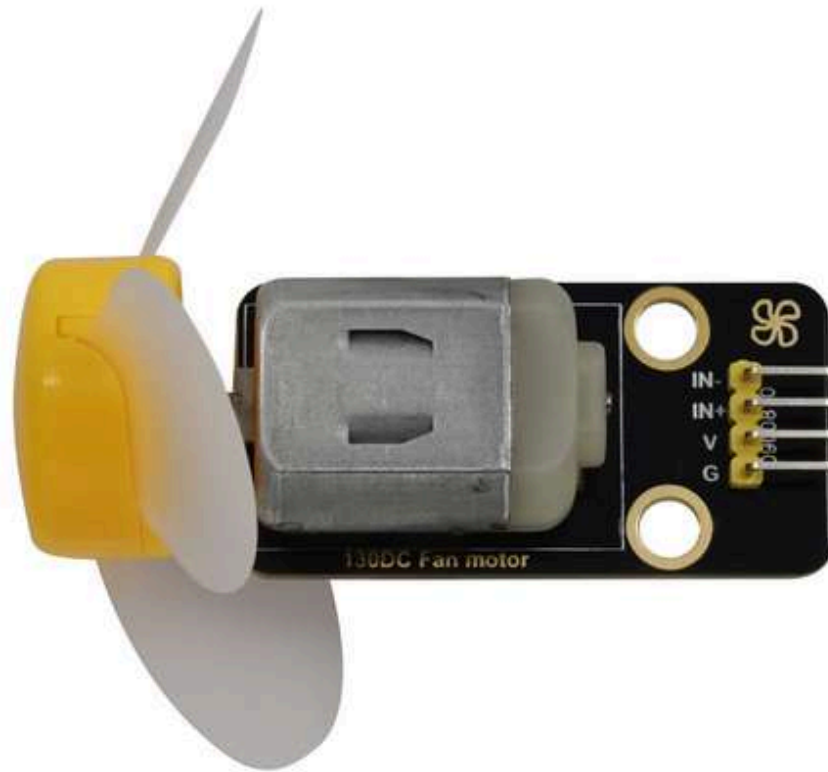


4.7.4 Modul ventilator

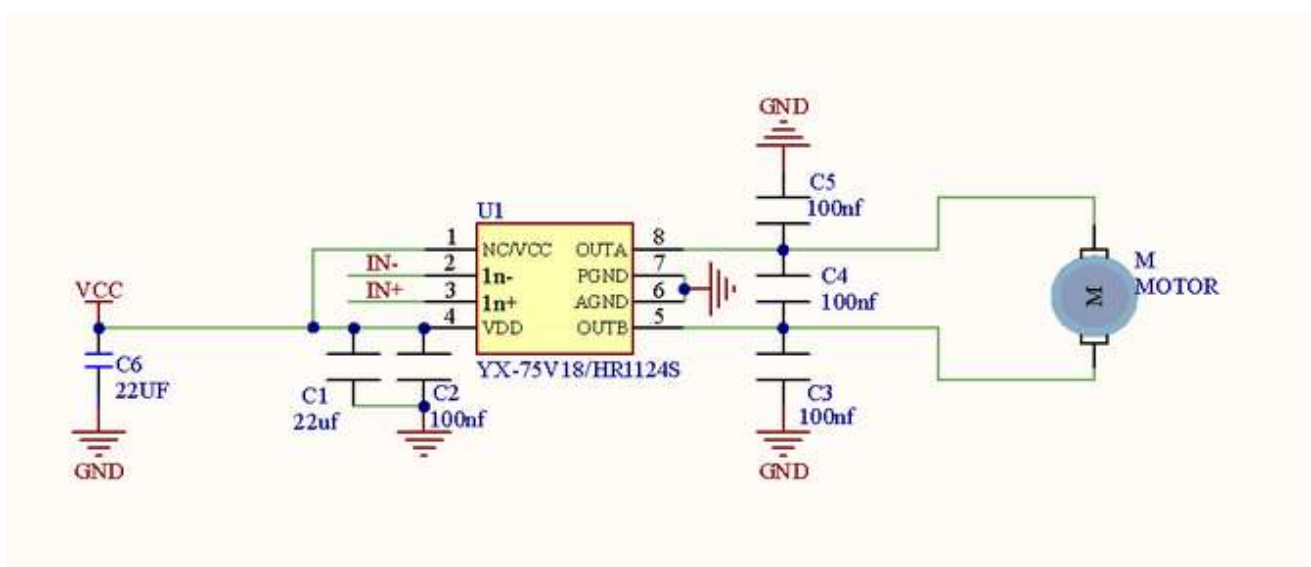
Descriere:

Motorul 130 este capabil să regleze viteza prin PWM. În acest proces, doi știfturi sunt trebuia conectat pentru control.

Modulul este potrivit pentru mai multe aplicații, cum ar fi căldura computerului disipare și producție industrială. În plus, este compact și ușor de instalat, ceea ce este foarte practic.



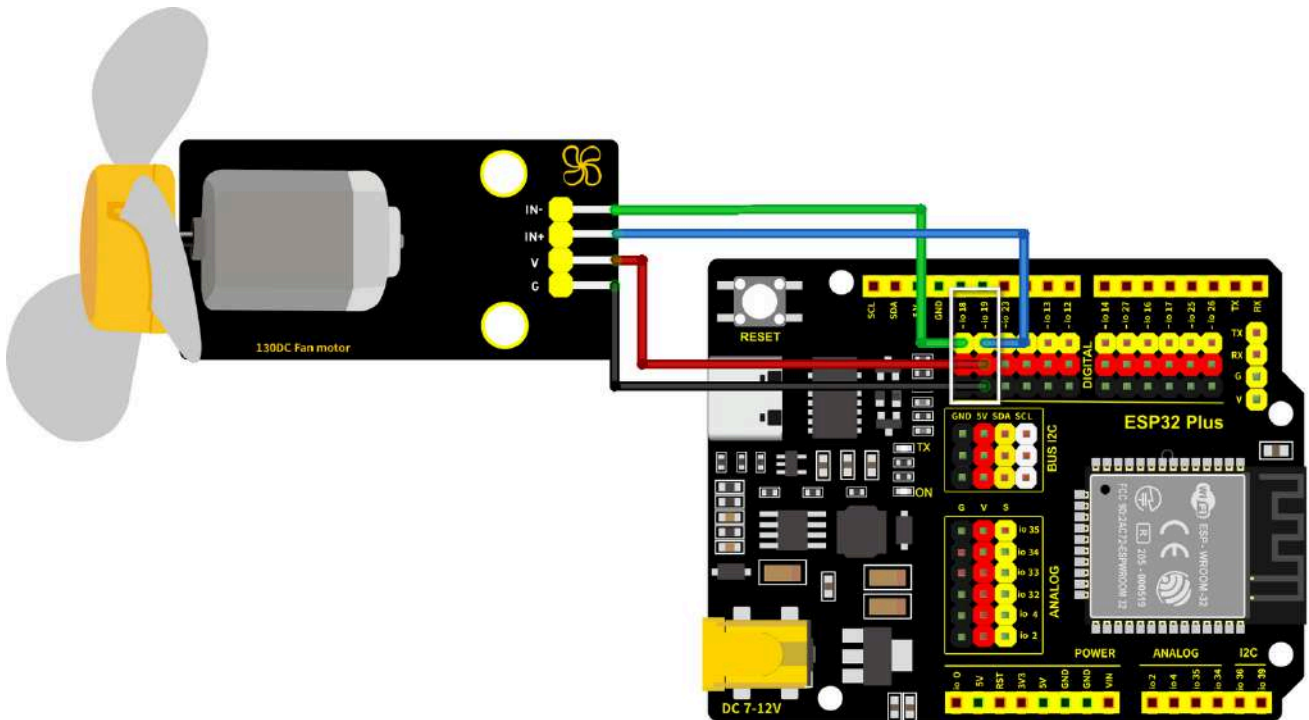
Diagramă schematică:



Schema:

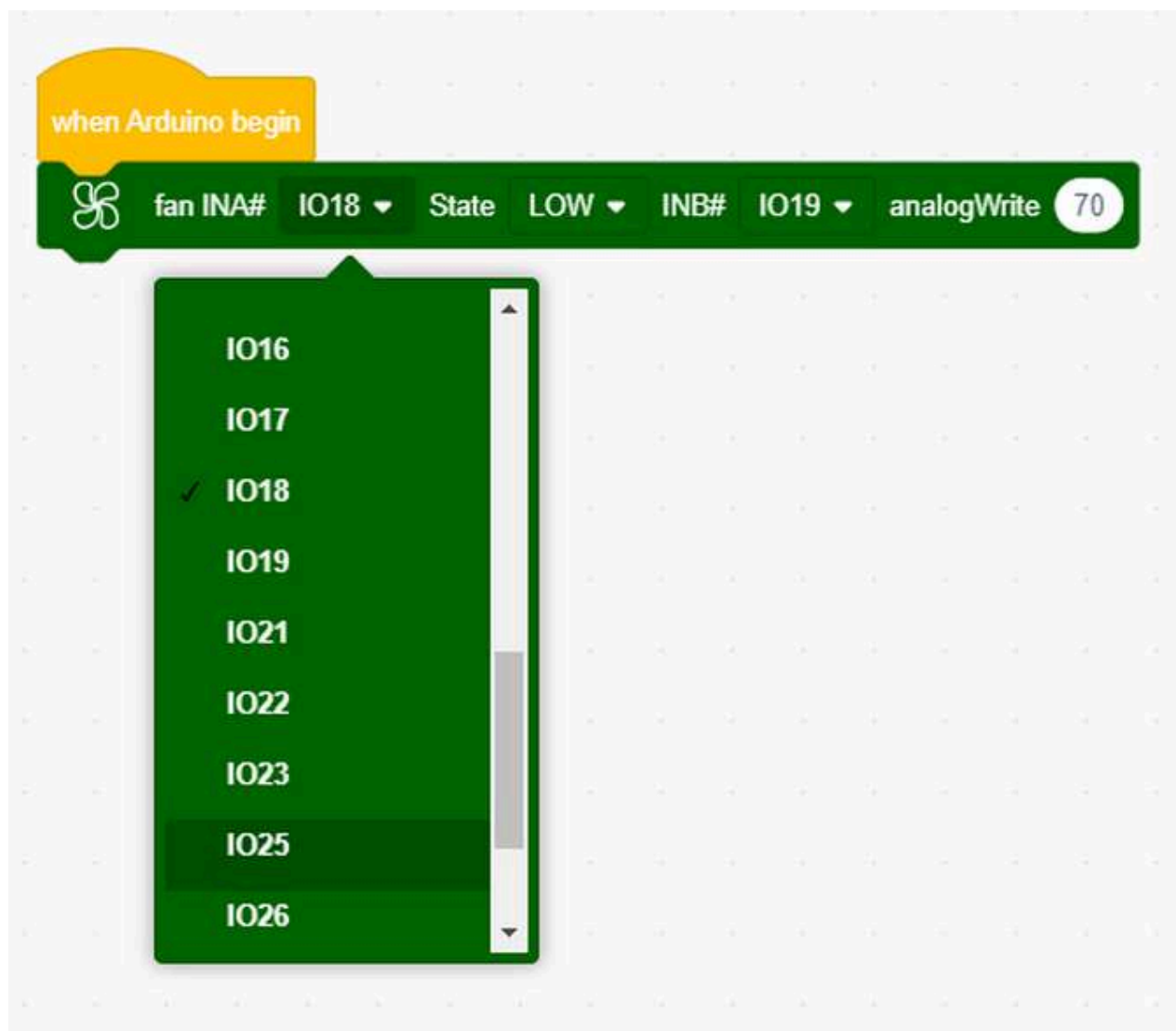
Conectați motorul la io18 și io19.

Atenție: Conectați galben la S (Semnal), roșu la V (Alimentare) și negru la GND. Nu le inversați!



Cod de testare:

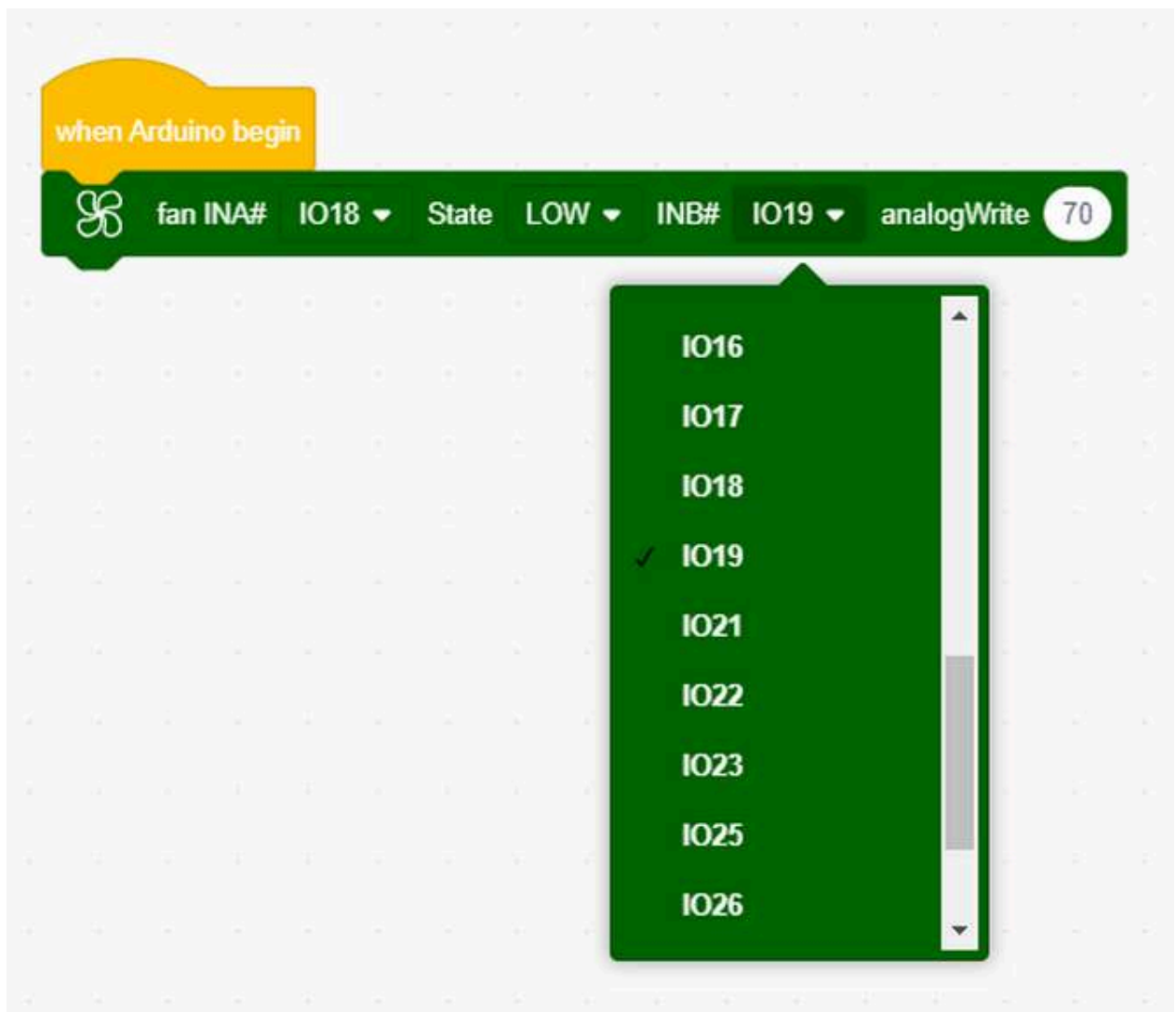
- Setări știftul ventilatorului INA



- Setezi starea nivelului de putere al INA, care determină rotația direcția ventilatorului.



- Setezi pinul ventilatorului INB.



- Setați ieșirea analogică la **INB**, care decide viteza de rotație.
 - Când INA este la mare, cu cât ieșirea analogică la INB este mai mică, Mai repede ventilatorul se va roti.
 - Când INA este la scăzut, cu cât este mai mare ieșirea analogică la INB, Mai repede ventilatorul se va roti.



Rezultatul testului:

Motorul 130 se rotește alternativ la stânga și la dreapta la fiecare 2 secunde



****NOTĂ:****

Există opriri intermitente în timpul schimbării direcțiilor de rotație. Ele preveniți un curent excesiv în momentul inversării. În caz contrar, un Resetarea forțată poate apărea din cauza sursei insuficiente de alimentare a consiliul de dezvoltare.

4.7.5 Sistem de control al temperaturii

Descriere:

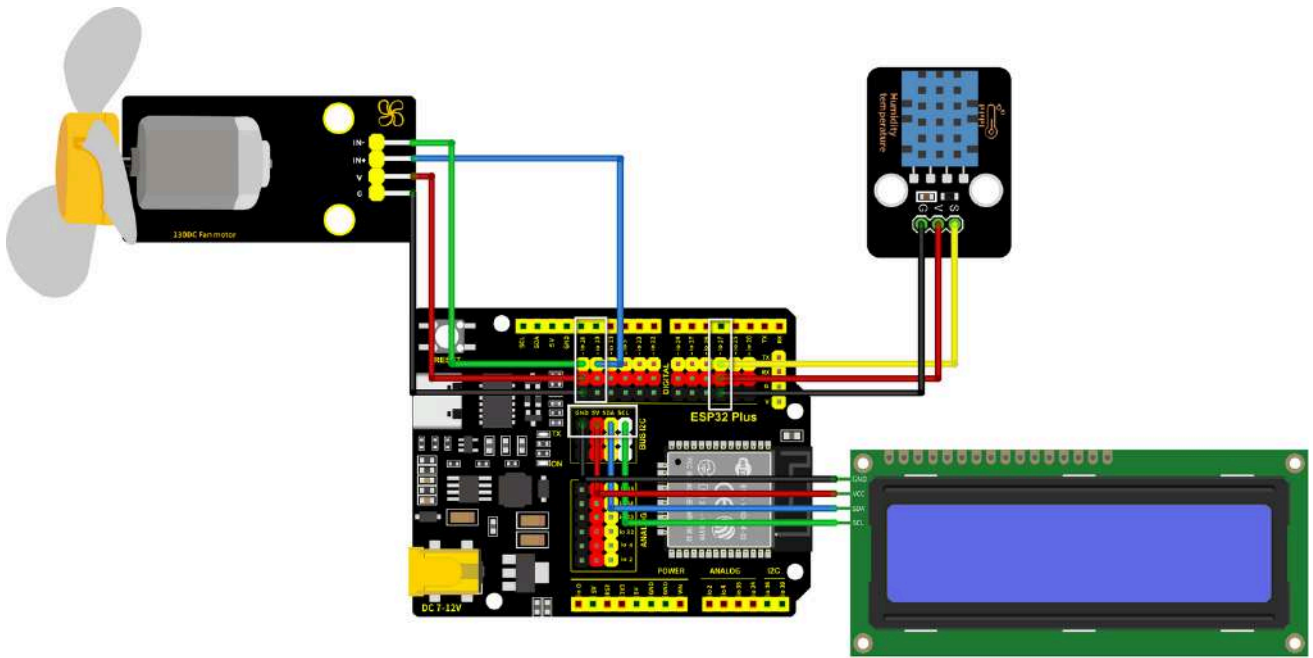
Aici, citim valoarea senzorului de temperatură și umiditate DHT11 prin comunicarea monobus, iar valorile vor fi afișate pe LCD. Dacă valorile depășesc pragul setat, ventilatorul se va aprinde pentru dezumidificarea și răcirea pentru a proteja animalele și plantele din fermă. În mod remarcabil, acest sistem este ușor de instalat cu mai multe funcții, cum ar fi controlul vitezei prin PWM și transmisia de date prin monobuz.

În general, este un sistem practic care ajută fermierii să monitorizeze și să controleze starea în timp real pentru a îmbunătăți eficiența producției.

Schema:

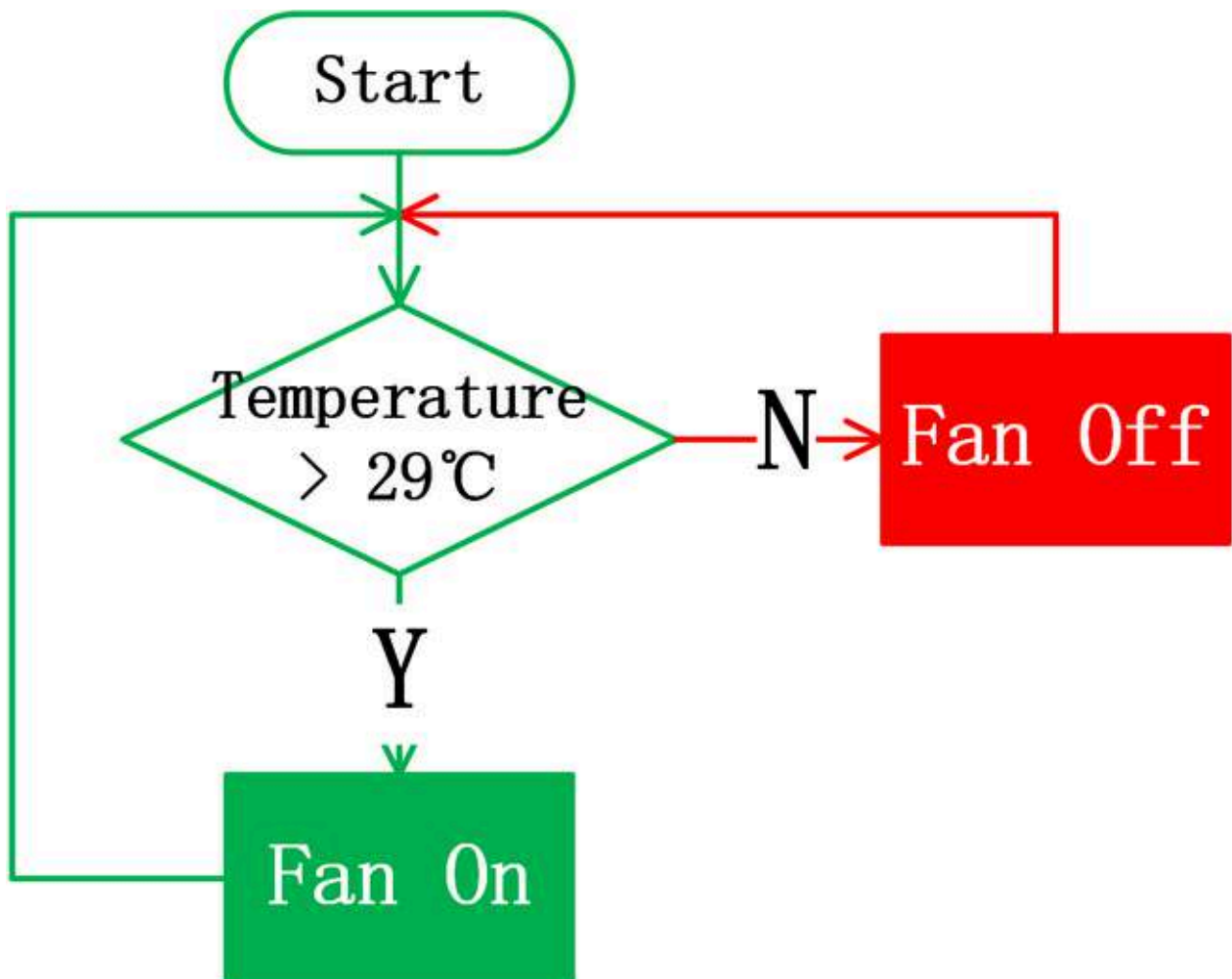
- Conectați senzorul de temperatură și umiditate la io17.
- Conectați motorul (ventilatorul) modue la io18 și io19
- Conectați LCD1602 la BUS I2C.

Atenție: Conectați galben la S (Semnal), roșu la V (Alimentare) și negru la GND. Nu le inversați!



Cod de testare:

Flux de cod:

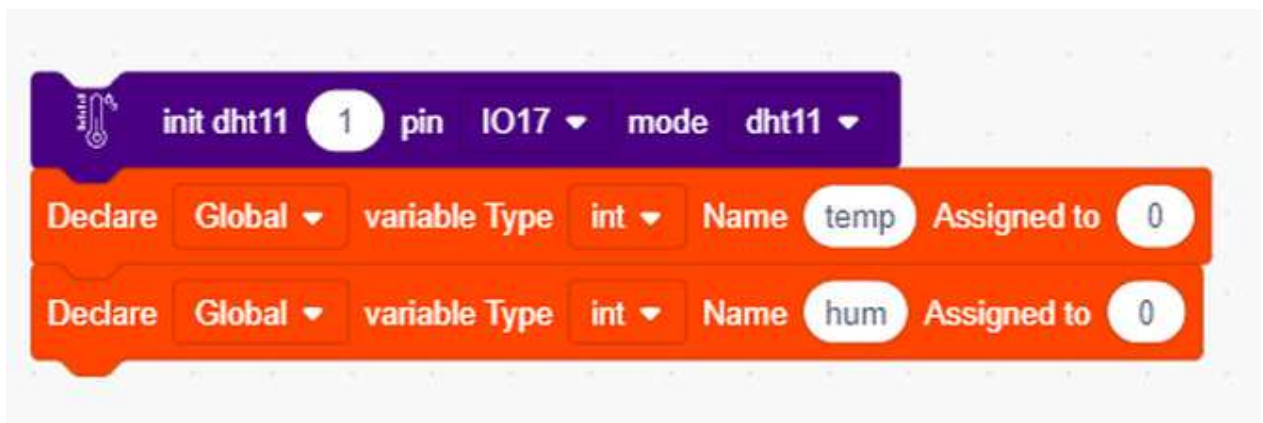


Cod:

- Inițializați ecranul LCD pentru a seta o adresă și ștergeți afișajul. Porniți-l Iluminare de fundal și setați poziția cursorului:



- Inițializați senzorul DHT11 și alegeți un pin corespunzător. Defini două variabile precum valorile temperaturii și umidității.



- În buclă, atribuiți valorile detectate celor două Variabile.



- Afișați valorile pe LCD.



- Determinați valoarea temperaturii și umidității. dacă temperatura este mai mare de 29° sau umiditatea depășește 80, ventilatorul se va roti.

```
if (variable temp > 29) or (variable hum > 80) then
  fan INA# IO18 State HIGH INB# IO19 analogWrite 70
else
  fan INA# IO18 State LOW INB# IO19 analogWrite 0
  wait 1 seconds
wait 0.5 seconds
```

The image shows a Scratch-style code block for controlling a fan. It starts with an 'if' block that checks two conditions: 'variable temp > 29' or 'variable hum > 80'. If either condition is true, the 'then' block executes, which is a 'fan' block with 'INA# IO18', 'State HIGH', 'INB# IO19', and 'analogWrite 70'. If the conditions are false, the 'else' block executes, which is a 'fan' block with 'INA# IO18', 'State LOW', 'INB# IO19', and 'analogWrite 0'. After the 'else' block, there is a 'wait 1 seconds' block. Finally, there is a 'wait 0.5 seconds' block at the bottom of the code block.

Cod complet:

when Arduino begin

init lcd I2C address 0x27

clear lcd

set lcd back light on

set lcd cursor position x: 0 y: 0

lcd print Temp:

set lcd cursor position x: 0 y: 1

lcd print Hum:

init dht11 1 pin IO17 mode dht11

Declare Global variable Type int Name temp Assigned to 0

Declare Global variable Type int Name hum Assigned to 0

forever

Set temp variable by dht11 1 read temperature

Set hum variable by dht11 1 read humidity

set lcd cursor position x: 5 y: 0

lcd print variable temp

set lcd cursor position x: 5 y: 1

lcd print variable hum

if variable temp > 29 or variable hum > 80 then

fan INA# IO18 State HIGH INB# IO19 analogWrite 70

else

fan INA# IO18 State LOW INB# IO19 analogWrite 0

wait 1 seconds

Ultima

wait 0.5 seconds

Rezultatul testului:

Când temperatura atinge 29°C, ventilatorul se va aprinde pentru a se disipa căldură. Când este mai mic de 29°C, ventilatorul se va opri (ventilatorul doar simulează disiparea căldurii, deci efectul nu este bun), ceea ce economisește energie pentru fermă.

4.7.6 ÎNTREBĂRI FRECVENTE

#Q: Senzorul de temperatură și umiditate este rezistent la apă?

R: Nu. Detectează temperatura și umiditatea ambiantă (în aer), astfel încât Vă rugăm să nu-l puneți în apă.

#Q: Placa ESP32 este resetată când ventilatorul se rotește.

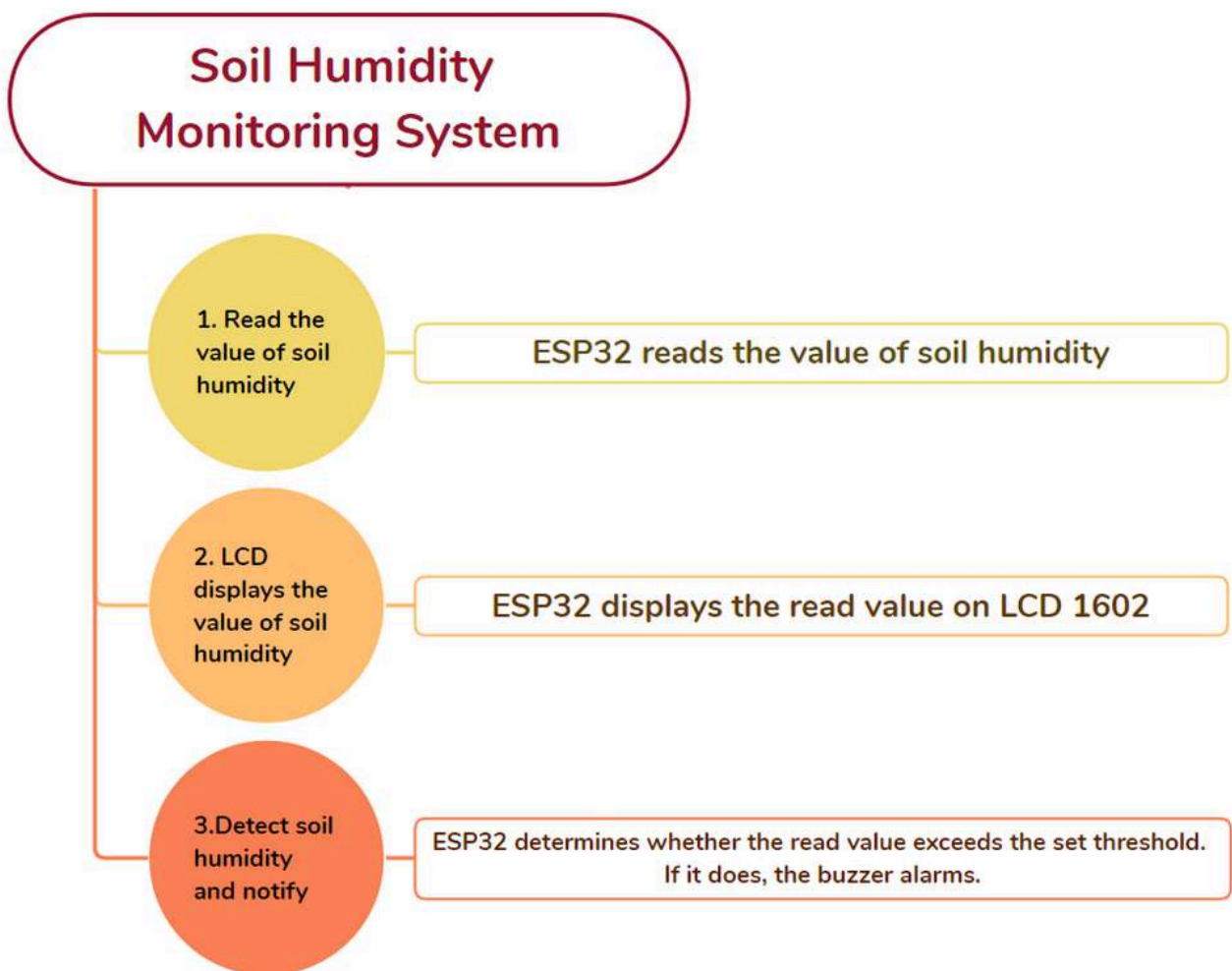
R: Când ventilatorul se rotește, este necesar mai mult curent decât alți senzori, prin urmare tensiunea și curentul pot fluctua în circuit. Mai ales la în momentul inversării ventilatorului, fluctuațiile pot fi prea mari, rezultând o resetare datorită tensiunii și curentului extrem de scăzute în dezvoltarea ESP32 scândură.

4.8 Proiect: Sistem de monitorizare a umidității solului

Fii atent! Nu revărsați apa din bazinele de plastic Experimente. Vărsarea apei pe alți senzori poate provoca nu numai un scurtmetraj circuite sau module pentru a fi în afara locului de muncă, dar și generarea de căldură și chiar explozie. Fii foarte atent! În special pentru utilizatorii mai tineri, vă rugăm operează cu părinții tăi. Pentru a garanta securitatea, vă rugăm să respectați instrucțiunile și reglementări de siguranță.



4.8.1 Diagrama de flux



4.8.2 Senzor de umiditate a solului

Descriere:

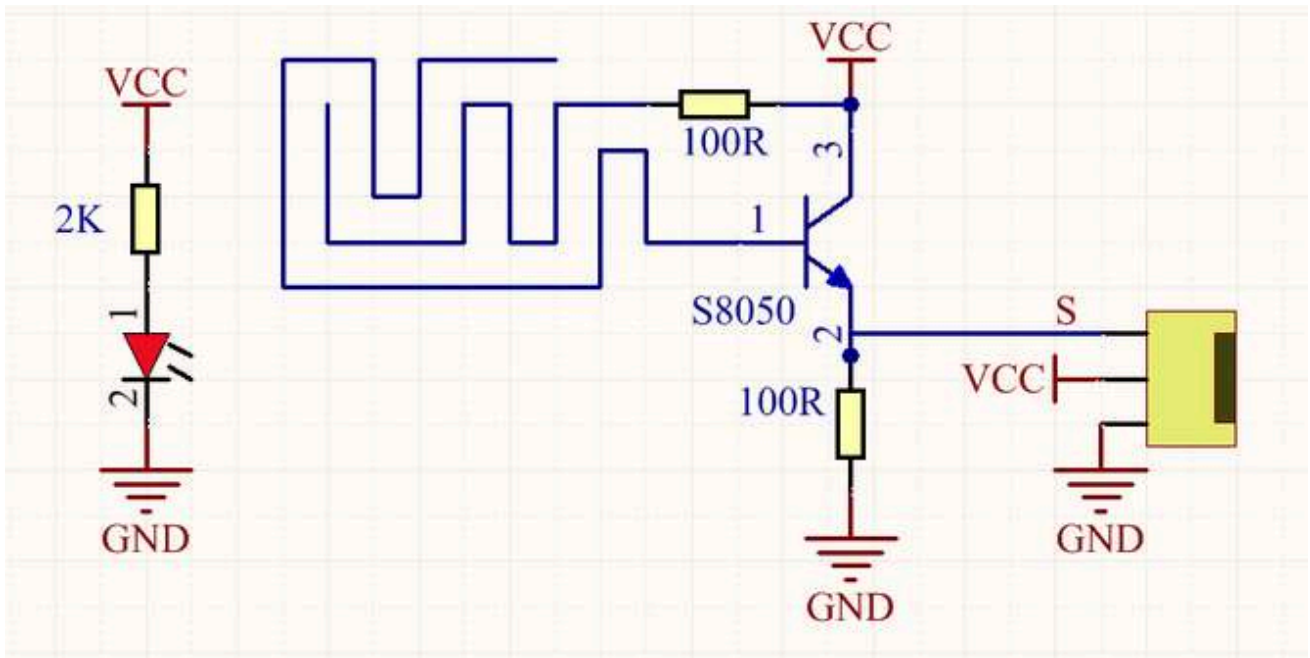
Senzorii de umiditate a solului sunt utilizați în principal pentru a măsura conținutul de apă în solul volumetric, monitorizează umiditatea solului, irigă culturile și protejează Păduri. Acest tip de senzor este integrat în irigarea agricolă de alimentare cu apă în mod regulat și eficient, care optimizează irigare pentru o cea mai bună creștere a plantelor.



Size: 63*20*8 mm

Weight: 2.5 g

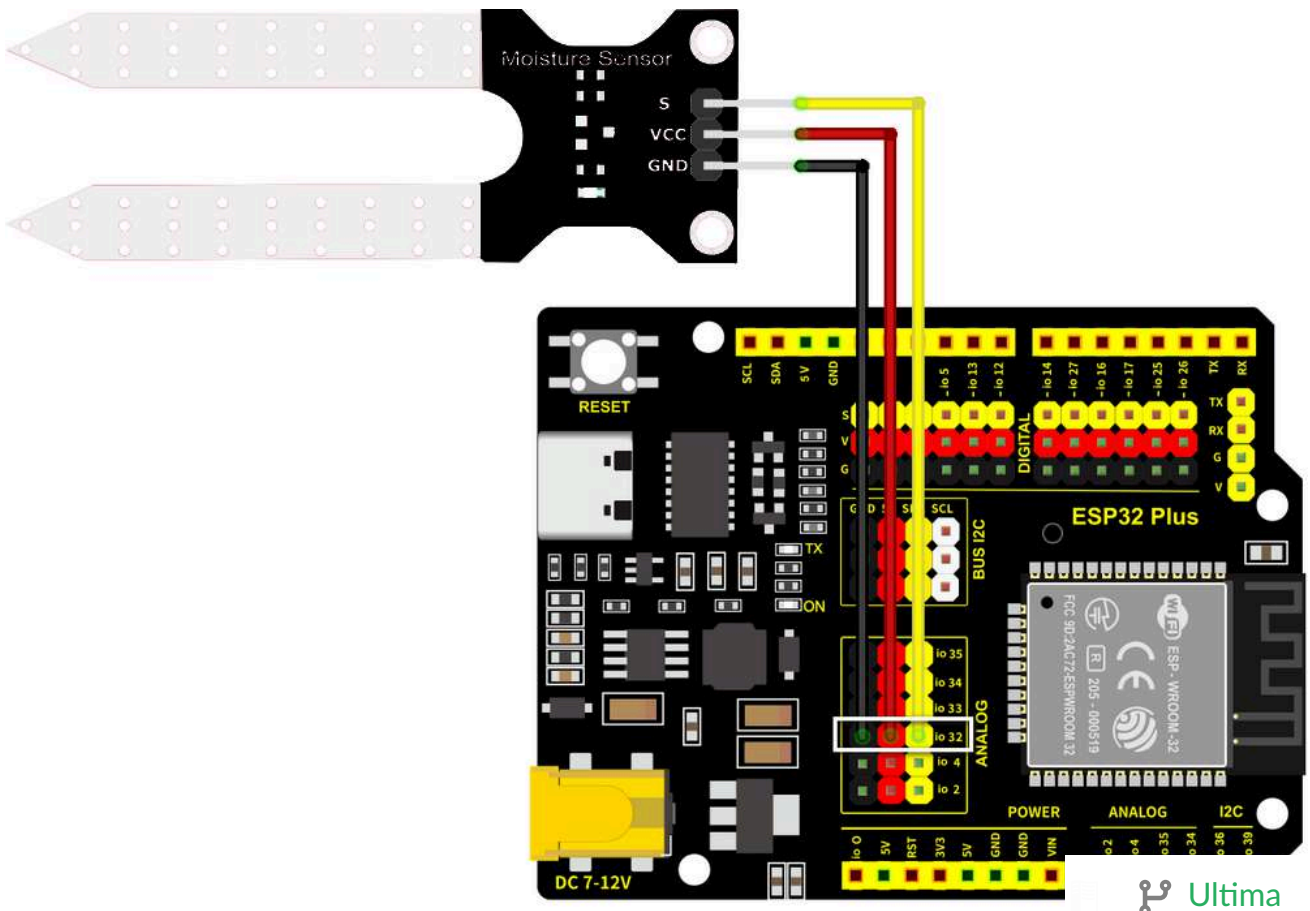
Diagramă schematică:



Schema:

Conectați senzorul de umiditate a solului la io32.

Atenție: Conectați galben la S (Semnal), roșu la V (Alimentare) și negru la GND. Nu le inversați!

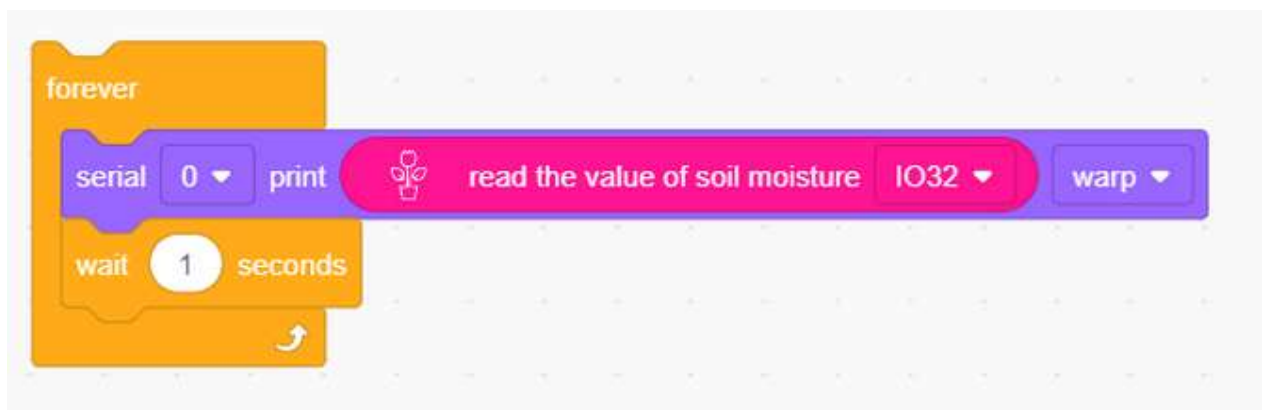


Cod de testare:

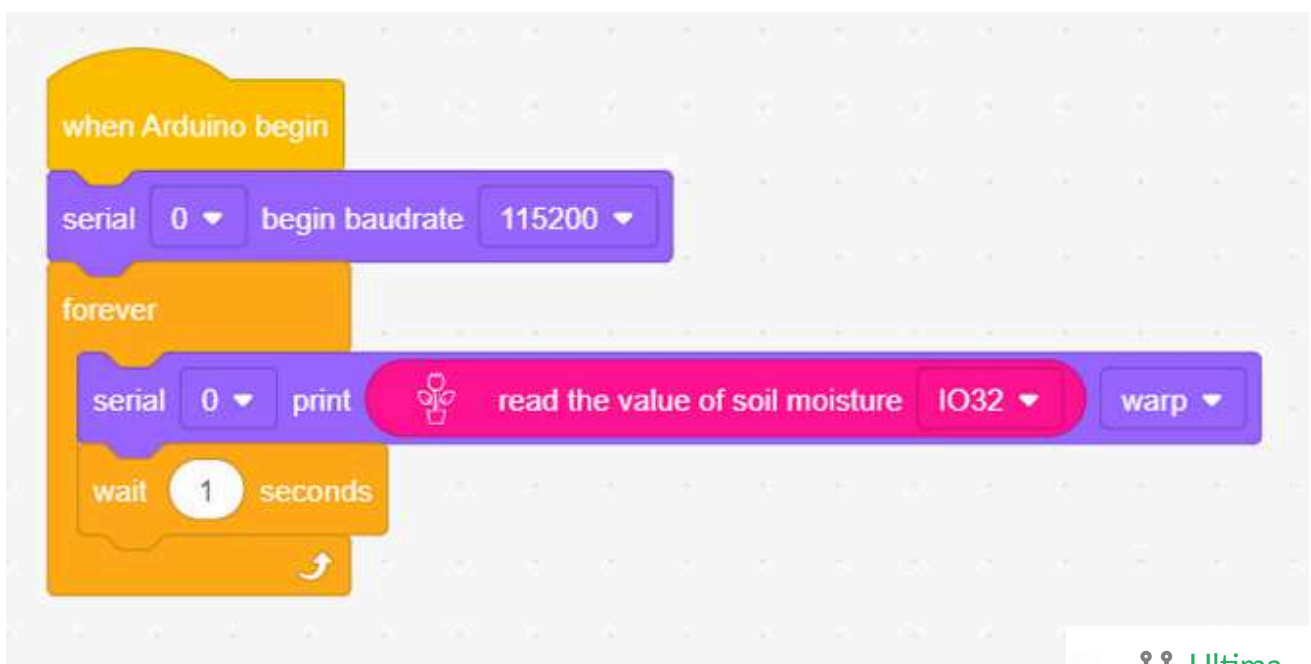
- Inițializați portul serial.



- Imprimați valoarea senzorului de citit.



Cod complet:



Rezultatul testului:

Deschideți monitorul serial.

Atingeți zona de detectare a senzorului cu un deget umed și Valoarea umidității detectată în prezent va fi imprimată pe monitor (interval: 0~4095).



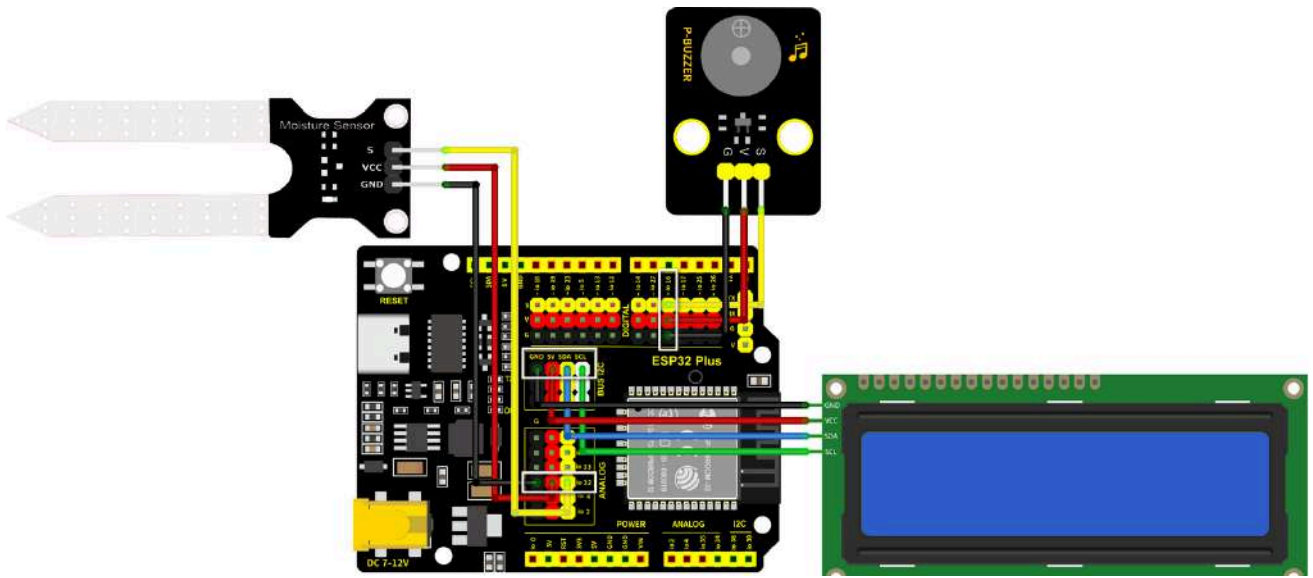
4.8.3 Sistemul de monitorizare a umidității solului

Adoptăm LCD1602 pentru a dezvălui valoarea în timp real a valorii umidității solului. Când valoarea este mai mică decât umiditatea minimă setată, soneria va emit sunet pentru a solicita fermierii de irigații.

Schema:

- Conectați senzorul de umiditate a solului la io32.
- Conectați soneria la io16.
- Conectați LCD1602 la BUS I2C.

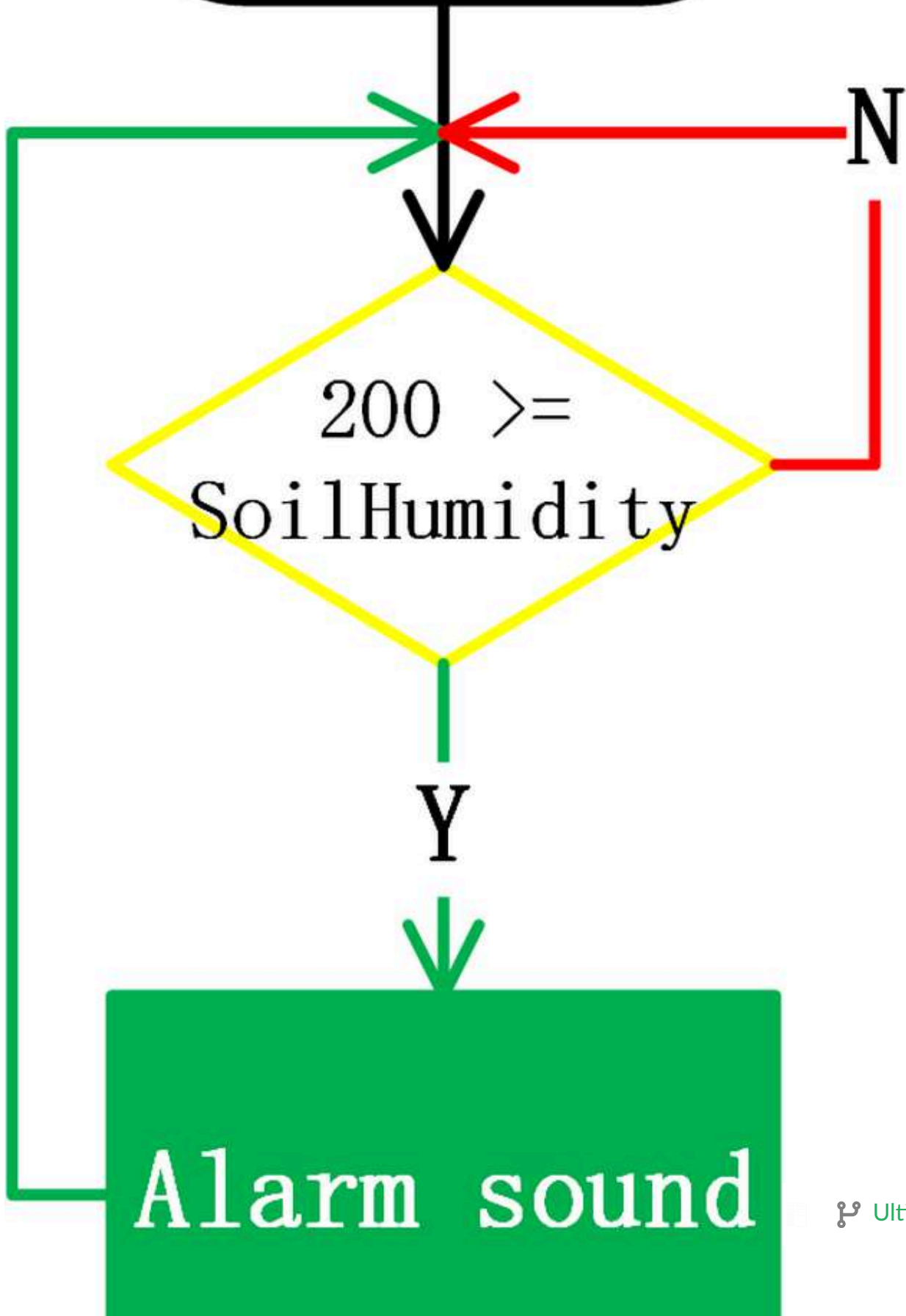
Atenție: Conectați galben la S (Semnal), roșu la V (Alimentare) și negru la GND. Nu le inversați!



Cod de testare:

Flux de cod:

Start



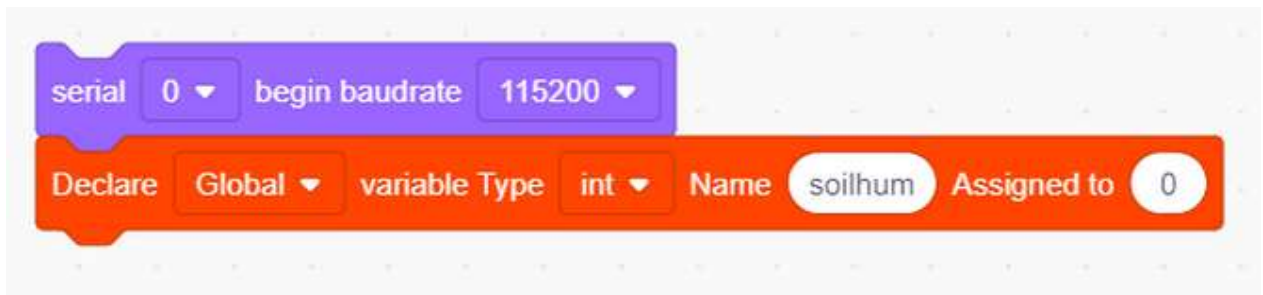


Cod:

- Inițializați ecranul LCD și ștergeți afișajul acestuia. Porniți lumina de fundal pentru a Respectați valoarea afișată.



- Inițializați portul serial și definiți o variabilă.



- Citiți valoarea umidității solului și atribuiți-o variabil. LCD arată valoarea.

Scratch code blocks for soil moisture reading and LCD display:

- Set `soilhum` variable by `read the value of soil moisture` (IO32)
- set lcd cursor position x: `0` y: `1`
- lcd print variable `soilhum`

- Determinați valoarea citită. Dacă este mai mic de 200, soneria va alarma.

Scratch code blocks for an if-then statement triggering an alarm:

- if `variable soilhum < 200` then
 - Tone PIN# `IO16` frequency `NOTE_C3` duration `131`
 - Tone PIN# `IO16` frequency `NOTE_G4` duration `131`
 - Tone PIN# `IO16` frequency `NOTE_D5` duration `131`
- NoTone `IO16`
- wait `1` seconds

Cod complet:

```
when Arduino begin
  init lcd I2C address 0x27
  clear lcd
  set lcd back light on
  set lcd cursor position x: 0 y: 0
  lcd print SoilHum:
  serial 0 begin baudrate 115200
  Declare Global variable Type int Name soilhum Assigned to 0
  forever
    Set soilhum variable by read the value of soil moisture IO32
    set lcd cursor position x: 0 y: 1
    lcd print variable soilhum
    if variable soilhum < 200 then
      Tone PIN# IO16 frequency NOTE_C3 duration 131
      Tone PIN# IO16 frequency NOTE_G4 duration 131
      Tone PIN# IO16 frequency NOTE_D5 duration 131
      NoTone IO16
    wait 1 seconds
```

Rezultatul testului:

Când valoarea detectată de senzorul de umiditate a solului este mai mică decât sciazi pragul, soneria emite sunet pentru alarmă.

4.8.4 ÎNTREBĂRI FRECVENTE

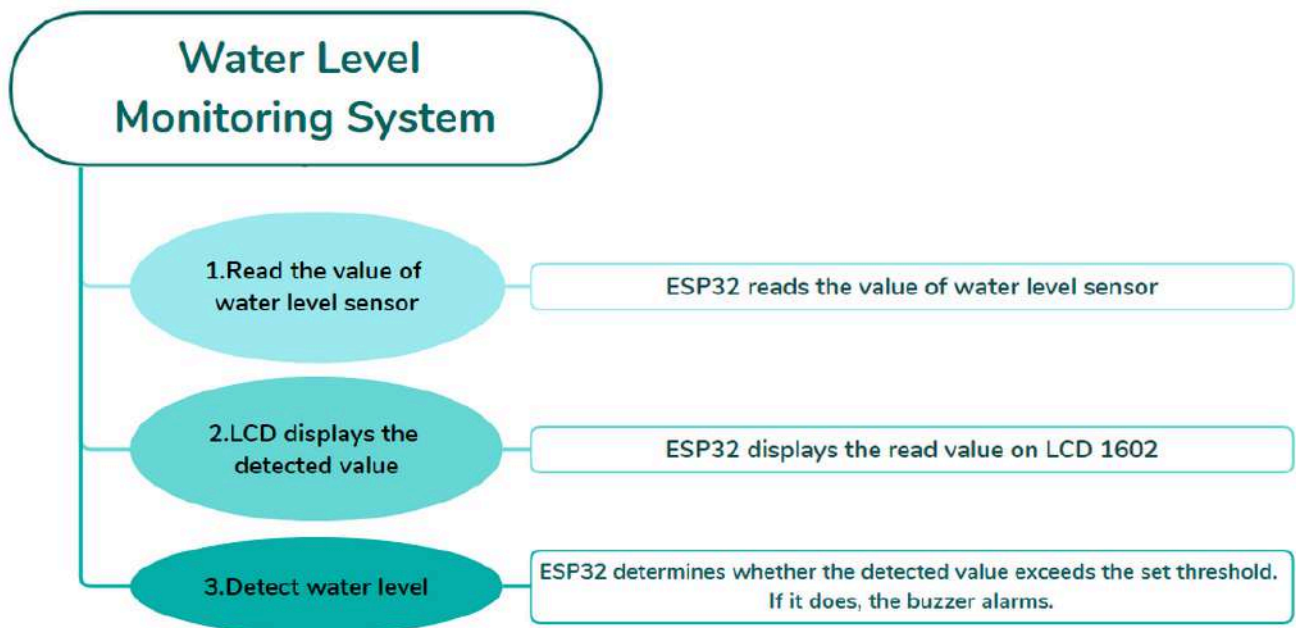
Î: Senzorul de umiditate a solului este rezistent la apă?

R: Cu excepția zonei de detectare, senzorul nu este impermeabil. Vărsarea apei pe altă zonă poate duce la un scurtcircuit.

4.9 Proiect: Sistem de monitorizare a nivelului apei



Fii atent! Nu revărsați apa din bazinele de plastic Experimente. Vărsarea apei pe alți senzori poate provoca nu numai un scurtmetraj pentru a perturba funcționarea normală, dar și generarea de căldură și chiar explozie. Fii foarte atent! În special pentru utilizatorii mai tineri, vă rugăm operează cu părinții tăi. Pentru a garanta securitatea, vă rugăm să respectați instrucțiunile și reglementări de siguranță.

4.9.1 Diagrama de flux



4.9.2 Senzor de nivel al apei

Descriere:

Senzorul de nivel al apei este ușor de utilizat, portabil și rentabil. El integrează  **Ultima**  paralele expuse pentru a măsura volumul apă și picături. Nu numai că senzorul este mai mic și mai inteligent decât alte detectoare de apă, dar dispune și:

- Tranziție lină între volumul de apă și volumul analogic;
- Flexibilitate puternică. Senzorul emite valori analogice de bază;
- Consum redus de energie și sensibilitate ridicată;
- Se conectează direct la microprocesoare sau circuite și este potrivit pentru diverse plăci de dezvoltare și controlere, cum ar fi Arduino controlere, microcomputere STC și AVR cu un singur cip.

Keyestudio Water Sensor

LOW POWER CONSUMPTION AND HIGH SENSITIVITY

- Size: 65mm x 20mm x 8mm
- Easy- to-use
- Portable
- Voltage: DC5V
- Current: < 20mA
- Working Humidity: 10%-90% without condensation
- N.W.:3.8g

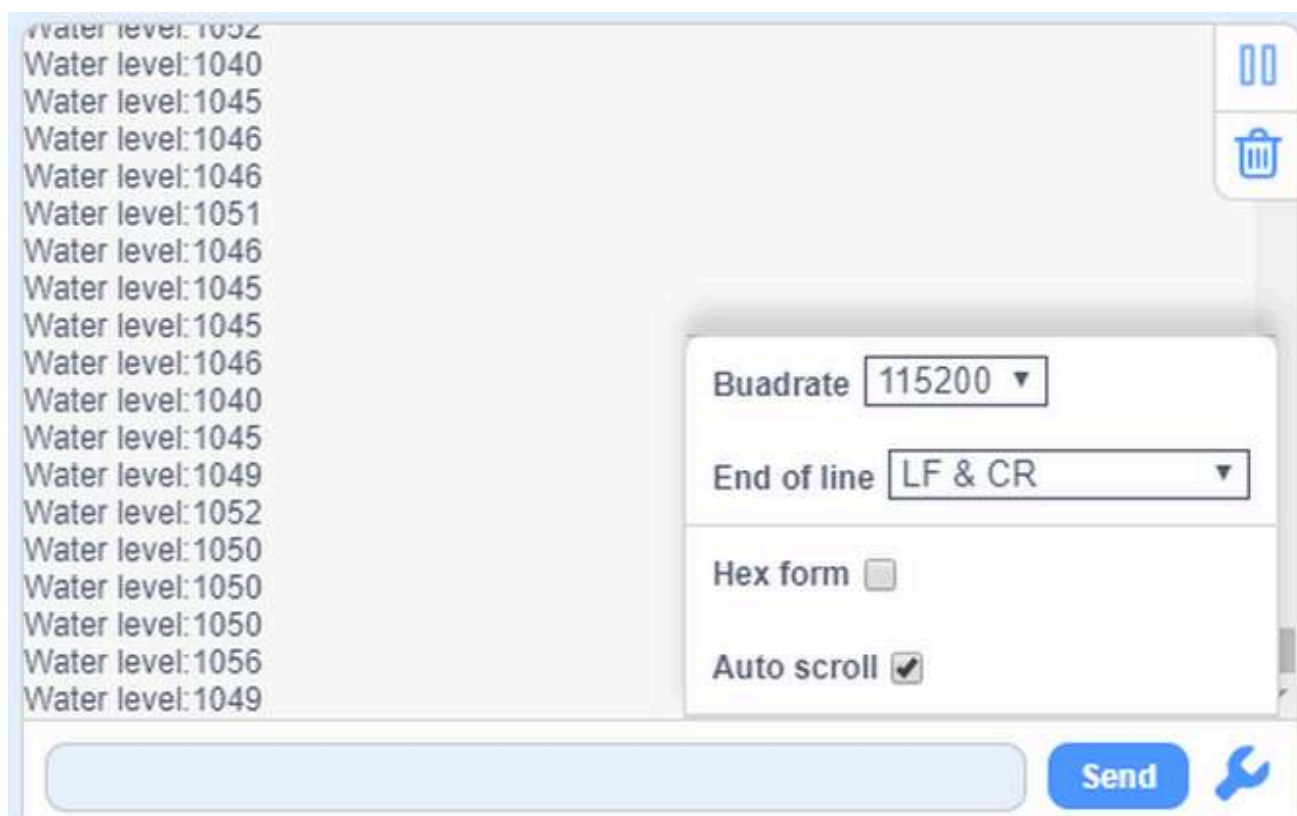


Schema:

Conectați senzorul de nivel al apei la io33.

Atenție: Conectați galben la S (Semnal), roșu la V (Alimentare) și negru la G
inversați!

Atingeți zona de detectare a senzorului cu un deget umed și Valoarea detectată în prezent va fi imprimată pe monitor (interval: 0~4095).



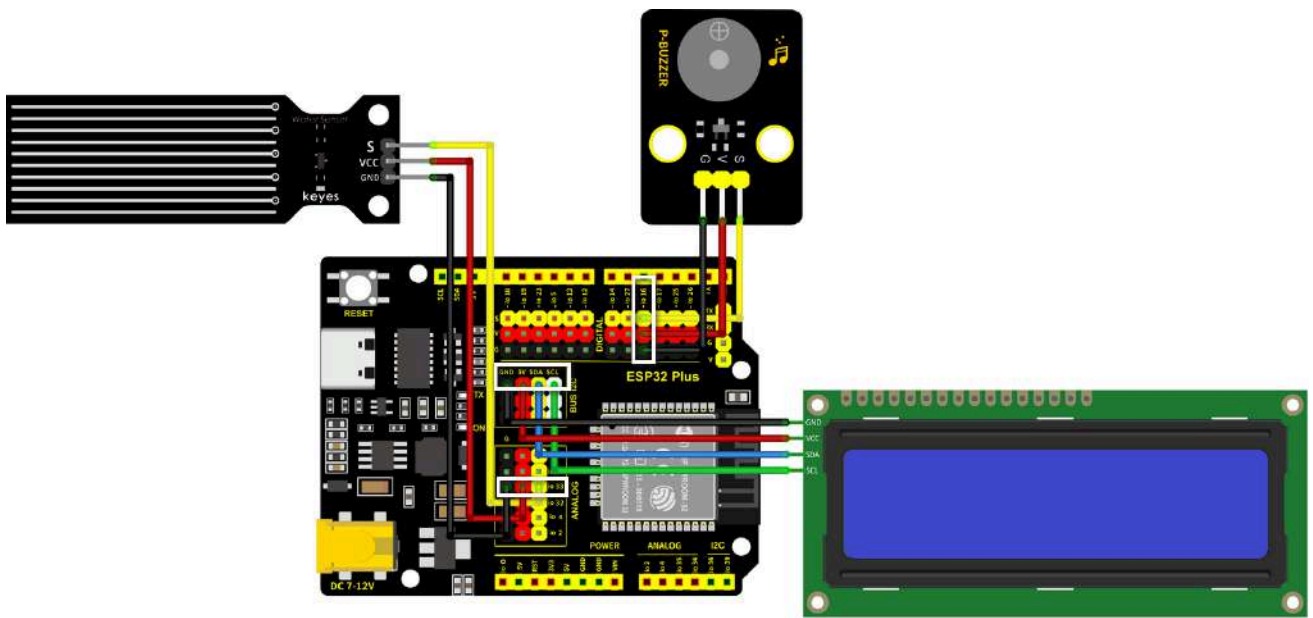
4.9.3 Sistemul de monitorizare a nivelului apei

Sistemul de monitorizare a nivelului apei supraveghează modificarea nivelului apei să clarifice problemele la timp și să ia măsuri pentru a evita dezastrea. Este utilizat pe scară largă în proiecte de conservare a apei, drenaj urban și monitorizarea mediului.

Schema:

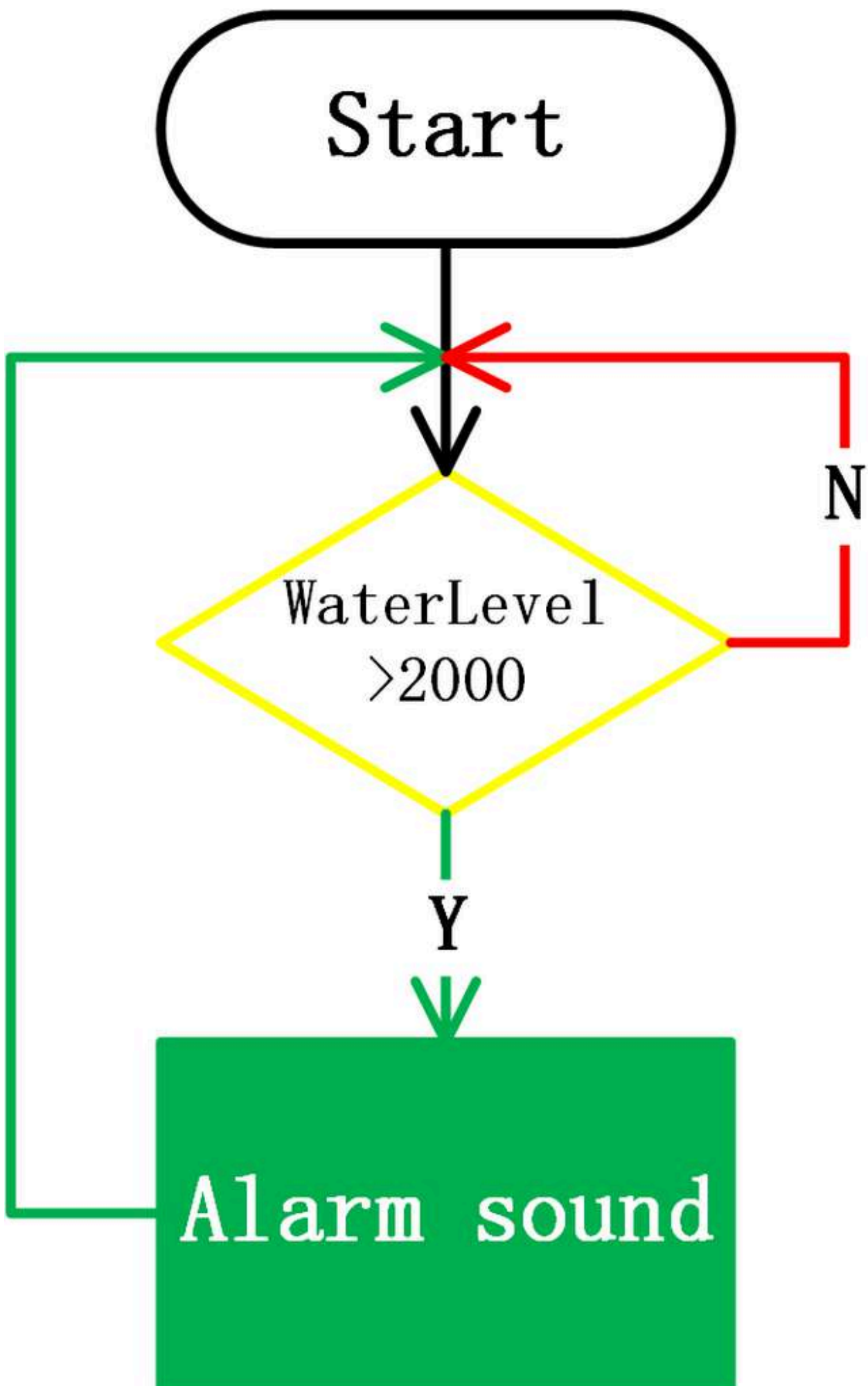
- Conectați senzorul de nivel al apei la io33.
- Conectați soneria la io16.
- Conectați LCD1602 la BUS I2C.

Atenție: Conectați galben la S (Semnal), roșu la V (Alimentare) și negru la GND. Nu le inversați!



Cod de testare:

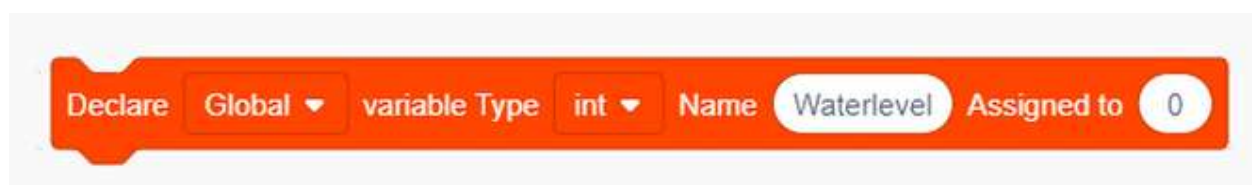
Flux de cod:



- Inițializați ecranul LCD și porniți lumina de fundal; Ștergeți toate afișajele și apoi imprimați nivelul apei.



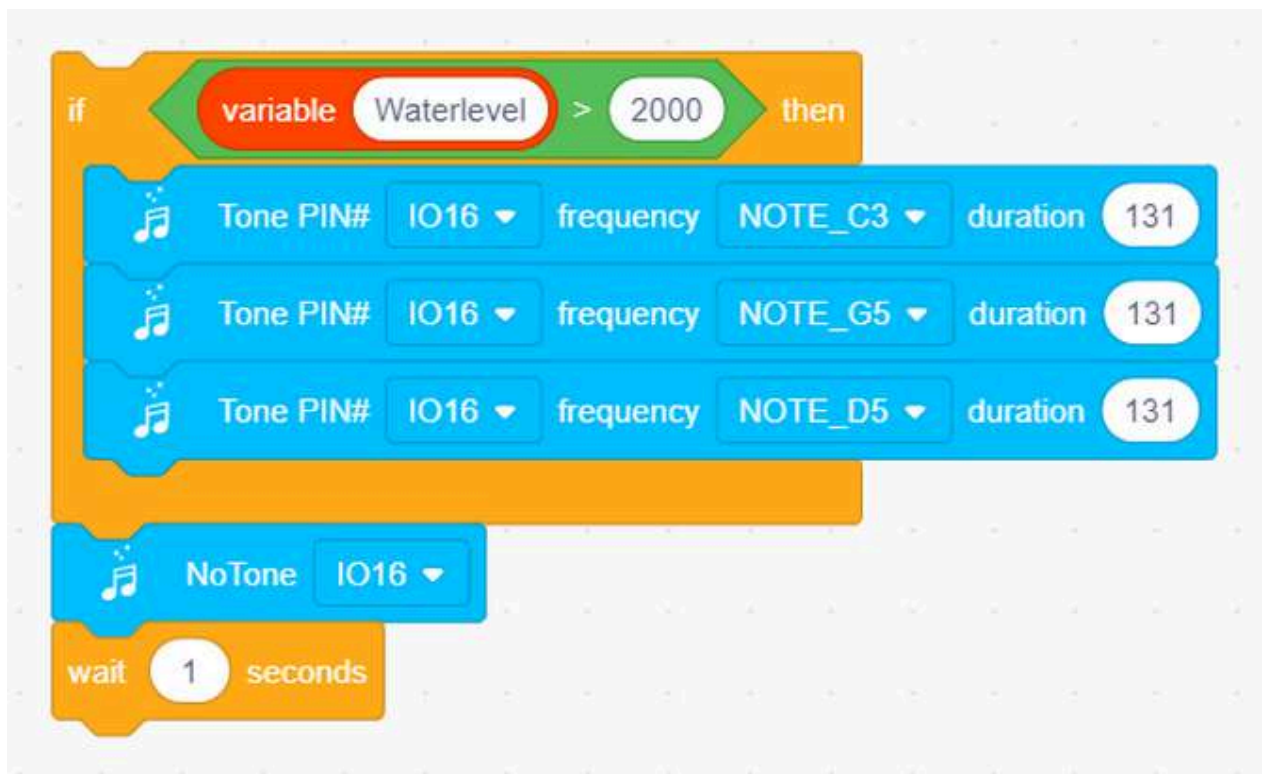
- Definiți o variabilă ca nivel de apă detectat.



- Citiți valoarea senzorului și afișați-o pe LCD.



- Determinați valoarea nivelului apei. Dacă este mai mare de 2000, Buzer va alarma.



Cod complet:

```
when Arduino begin
  init lcd I2C address 0x27
  set lcd back light on
  clear lcd
  set lcd cursor position x: 0 y: 0
  lcd print Water Level:

  Declare Global variable Type int Name Waterlevel Assigned to 0

  forever
    Set Waterlevel variable by read the value of water level IO33
    set lcd cursor position x: 0 y: 1
    lcd print variable Waterlevel

    if variable Waterlevel > 2000 then
      Tone PIN# IO16 frequency NOTE_C3 duration 131
      Tone PIN# IO16 frequency NOTE_G5 duration 131
      Tone PIN# IO16 frequency NOTE_D5 duration 131
    else
      NoTone IO16
    end if
    wait 1 seconds
```

Rezultatul testului:

LCD afișează valoarea în timp real a nivelului apei. În experiment, am Acoperiți zona de detectare cu apă pentru a stimula nivelul apei. Când Valoarea detectată depășește valoarea setată, soneria începe să emită alarmă.

4.9.4 ÎNTREBĂRI FRECVENTE

Î: Senzorul de nivel al apei este rezistent la apă?

R: Cu excepția zonei de detectare, senzorul nu este impermeabil. Vărsarea apei pe altă zonă poate duce la un scurtcircuit.

4.10 Proiect: Sistem de irigare automată

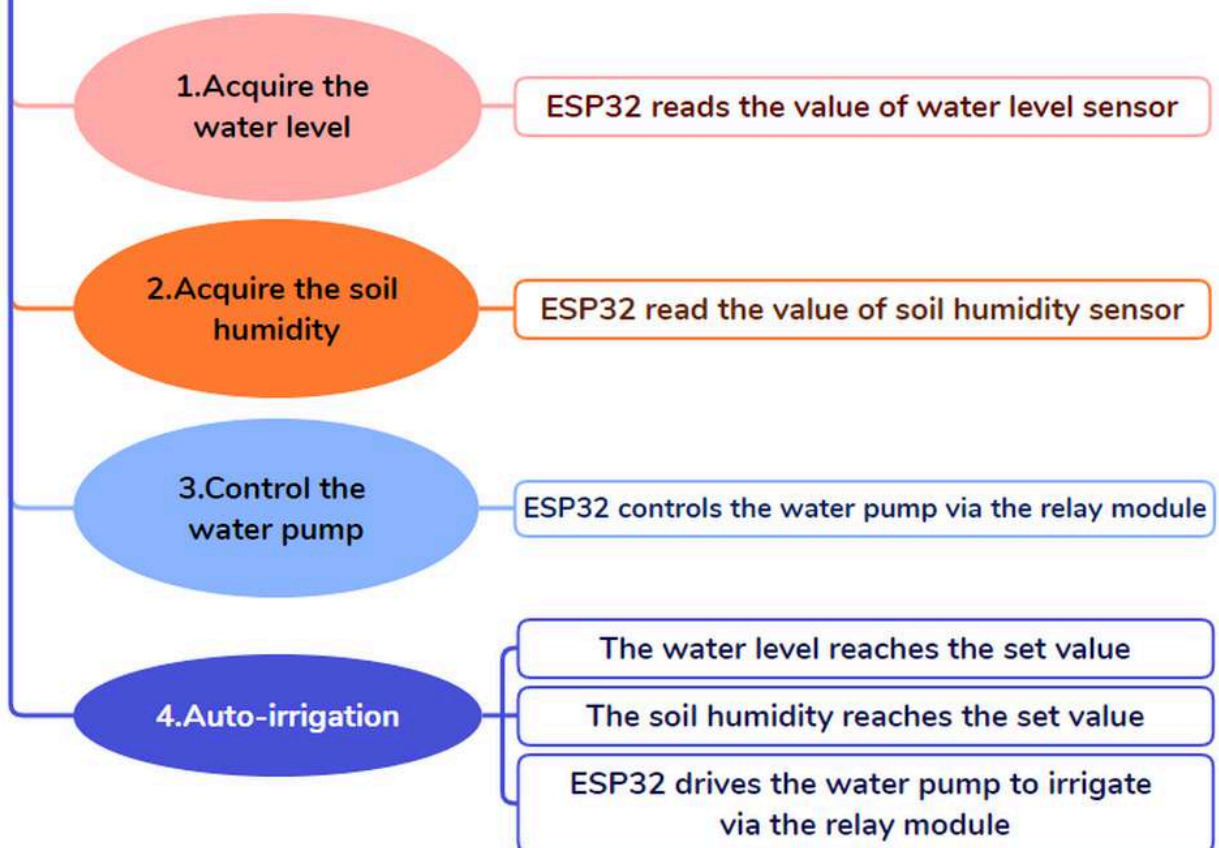
Fii atent! Nu revărsați apa din bazinele de plastic Experimente. Vărsarea apei pe alți senzori poate provoca nu numai un scurtmetraj pentru a perturba funcționarea normală, dar și generarea de căldură și chiar explozie. Fii foarte atent! În special pentru utilizatorii mai tineri, vă rugăm operează cu părinții tăi. Pentru a garanta securitatea, vă rugăm să respectați instrucțiunile și reglementări de siguranță.

În acest proiect, stimulăm irigarea printr-o pompă de apă controlată de un modul de releu. În plus, determinăm și dacă există apă în piscină prin senzorul de nivel al apei și detectează umiditatea solului în funcție de sol senzor de umiditate. În acest fel, sistemul va fi mai inteligent în controlul pompei de apă.



4.10.1 Diagrama de flux

Auto-Irrigation System



4.10.2 Sistem de pompare a apei

Descriere:

În acest experiment, folosim placa de dezvoltare ESP32 pentru a porni/opri pompă de apă printr-un modul de releu. O pompă ridică apa și transportă lichide, și de obicei este combinat cu un modul de releu în uz.

Aici, conectăm modulul releu și pompa la placa ESP32 și pentru a porni sau opri de la distanță pompa prin comutarea stării de releu. Pentru cum, determinăm starea releului în funcție de valoarea de ieșire a modulului sau un timp prestabilit.

Modul releu:

În utilizare, este adesea utilizat în gestionarea tensiunii înalte și a sarcinii curent, să zicem, motoare, senzori de curent mare și lumini de mare putere.



- **Deschis în mod normal (NO):** Acest pin este în mod normal deschis, cu excepția cazului în care un semnal este recepționat de știftul de semnal al releului. Prin urmare, știfturile comune sunt deconectat prin pinul NC și conectat prin pinul NO.
- **Contact comun (COM):** Acest pin se conectează la alte module, pentru Exemplu, pompa de apă.
 - Pompă de apă:

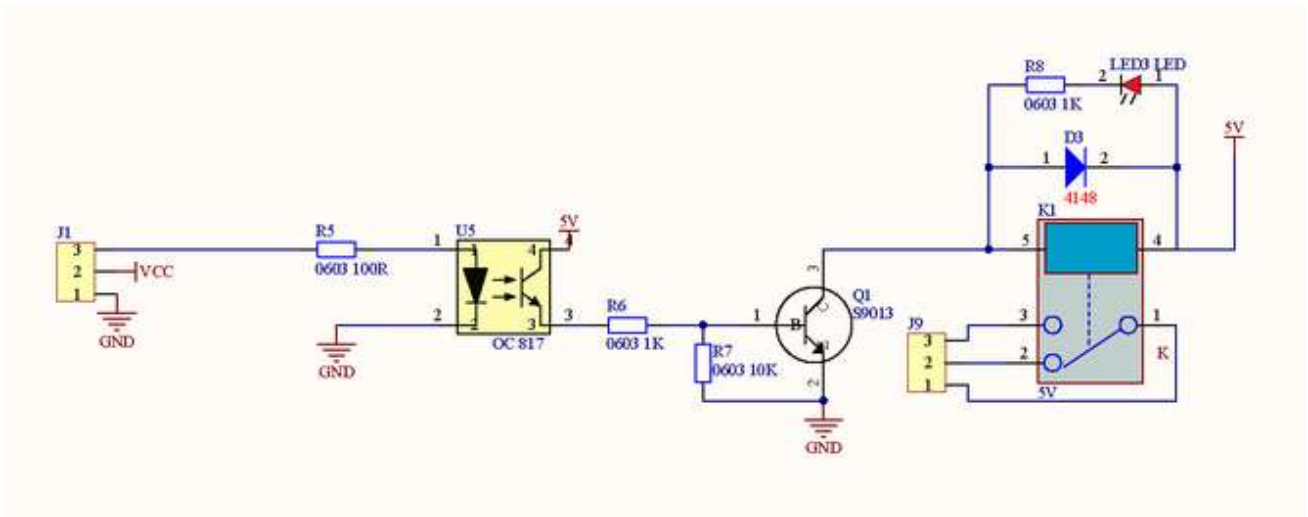


- **Închis în mod normal (NC):** Știftul NC este legat de pinul COM pentru a forma un circuit închis. Folosește placa ESP32 pentru a controla închiderea și deconectarea modulului releu.

Parametrii:

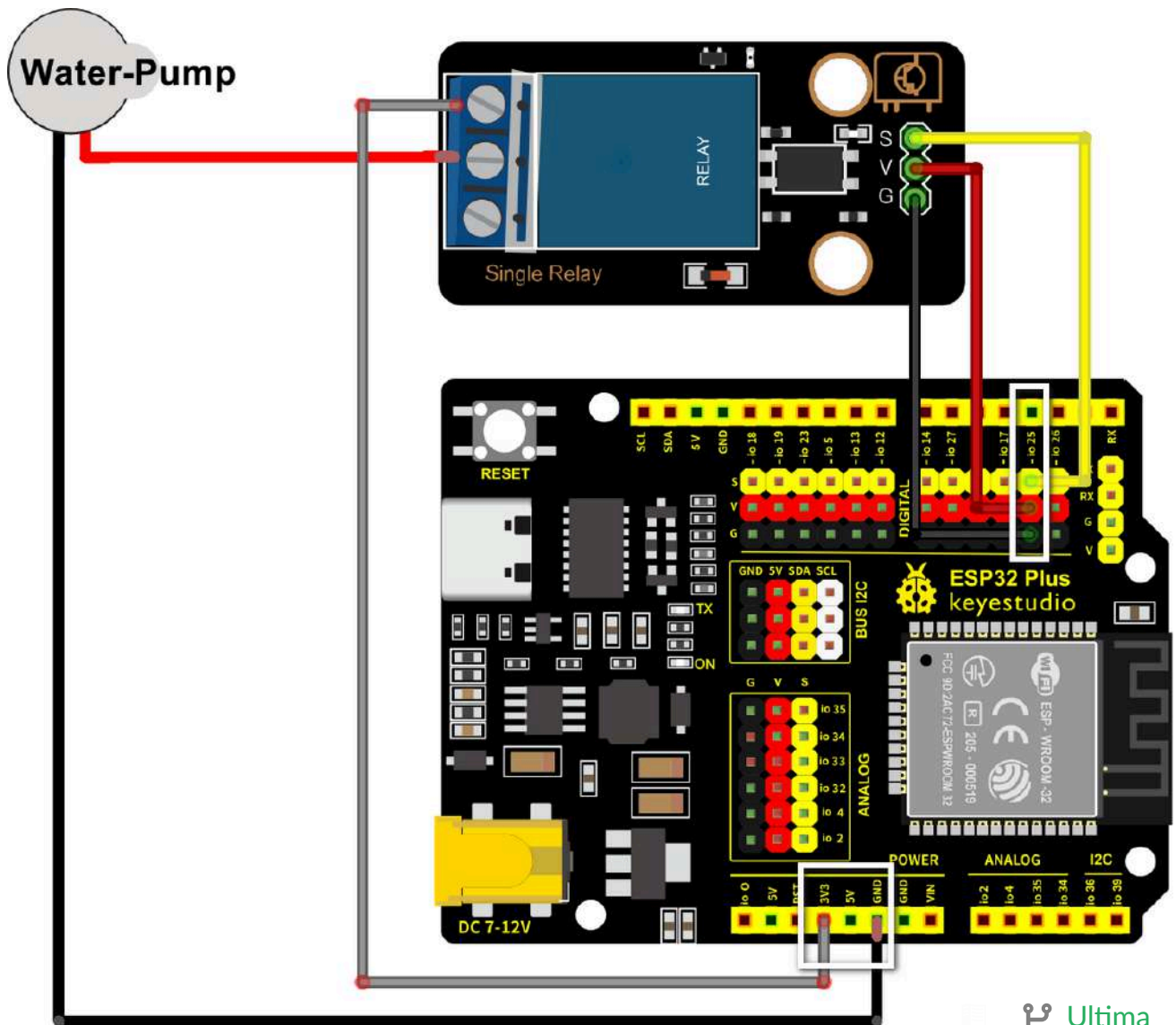
- Tensiune de putere: 5V
- Curent static: 2mA
- Tensiune maximă de contact: 250VAC/30VDC
- Curent maxim: 10A

Diagramă schematică:



Schema:

Atenție: Conectați galben la S (Semnal), roșu la V (Alimentare) și negru la GND. Nu le inversați!



Cod de testare:



Rezultatul testului:

După încărcarea codului, dispozitivul va pompa apă o dată.

În acest experiment, pompa de apă este automatizată, reducând timpul și de operare manuală și de îmbunătățire a eficienței. Prin urmare, acest Sistem de pompare a apei este utilizat în mod sălbatic în producția agricolă și în apă tratament.

4.10.3 Sistem de irigare automată

Descriere:

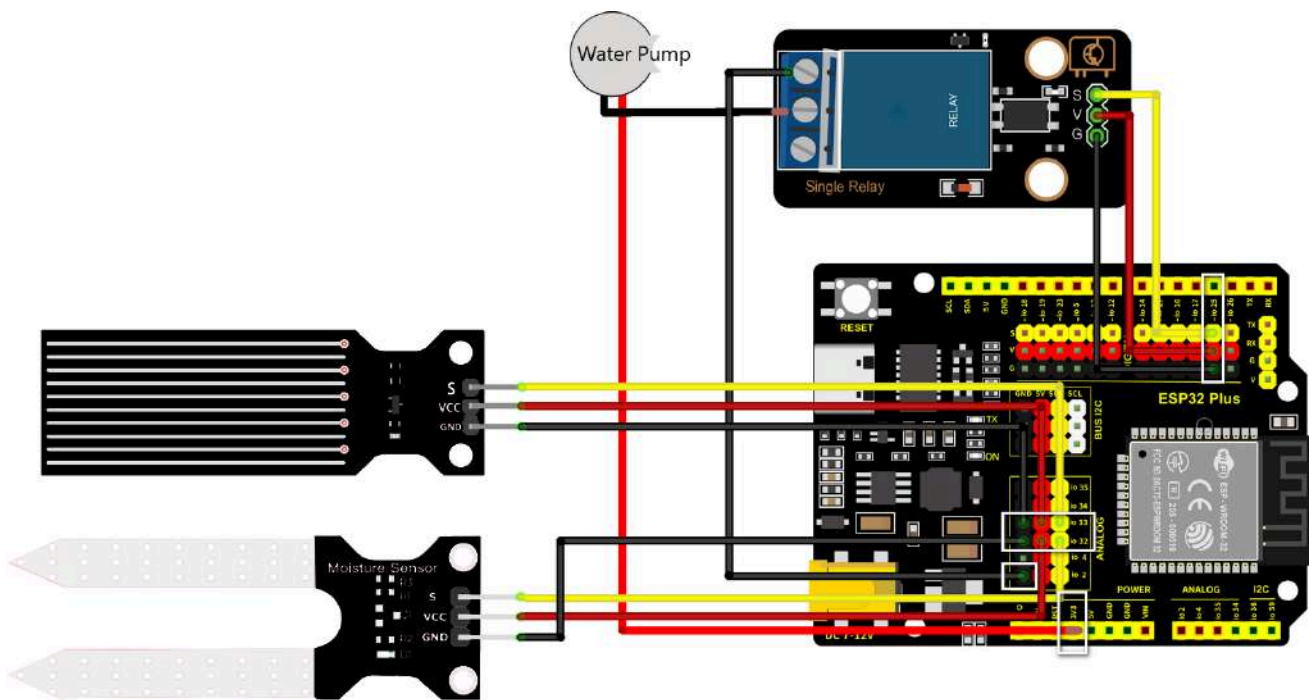
În acest experiment, implementăm un sistem inteligent de irigare printr-un sol senzor de umiditate, un senzor de nivel al apei, un modul de releu și o pompă de apă. Conectăm cei doi senzori de pe placa de dezvoltare ESP32 și programăm la Citiți valorile lor de ieșire pentru a controla releul și pompa de apă.

Dacă solul este foarte uscat, releul se va porni pentru a controla apa pompă pentru irigarea plantelor; Și dacă nivelul apei este prea scăzut, apa pompa nu va putea funcționa, iar soneria va alarma. În acest fel, udarea plantelor și controlul nivelului apei sunt automatizate, ceea ce crește eficiența producției și reduce timpul și eforturile de Operațiuni.

Schema:

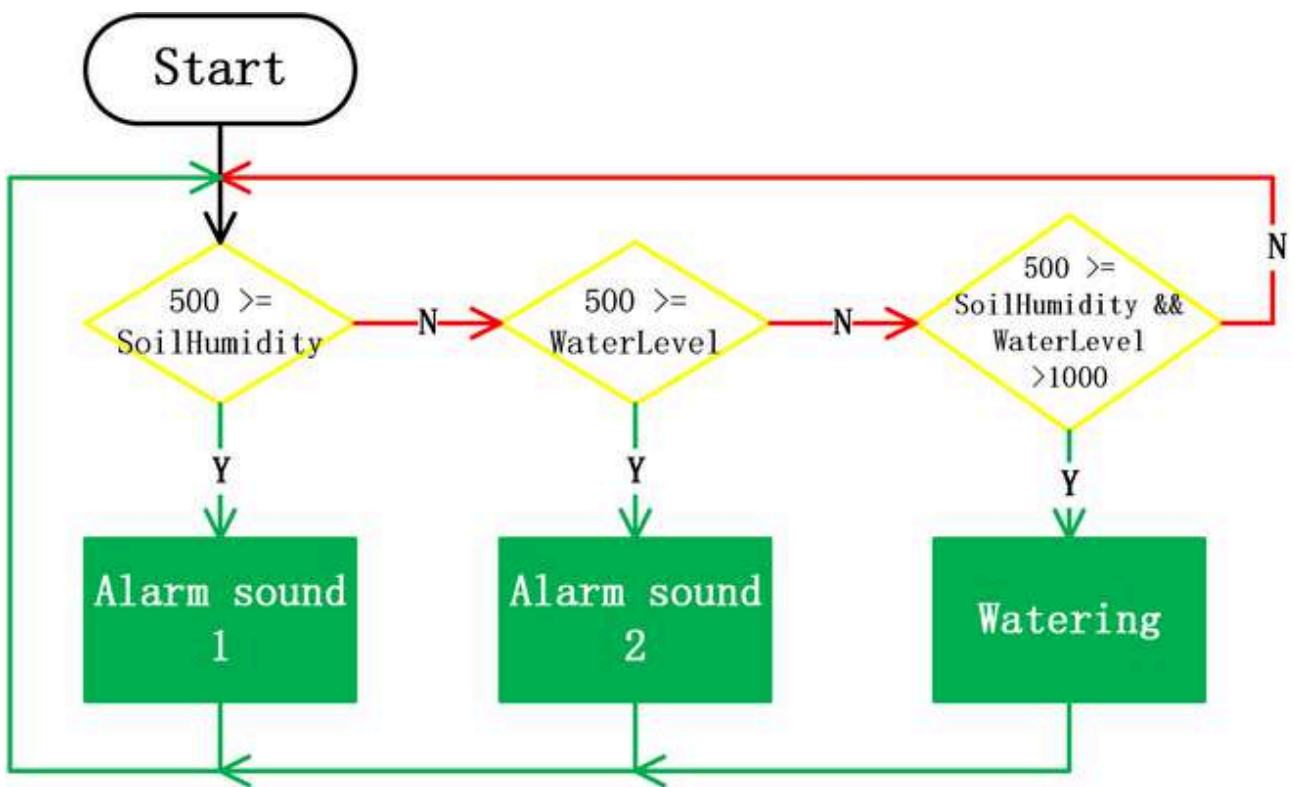
- Conectați modul releu la io25; conectați-i pinul NC la GND(negru) la io2.
- Pompă de apă:
 - Conectați firul roșu la POWER 3V3 al plăcii
 - Conectați firul negru (GND) la pinul COM al releului
- Conectați senzorul de umiditate a solului la io32
- Conectați senzorul de nivel al apei la io33

Atenție: Conectați galben la S (Semnal), roșu la V (Alimentare) și negru la GND. Nu le inversați!



Cod de testare:

Flux de cod:



Cod:

- Inițializați și ștergeți LCD-ul, porniți lumina de fundal LCD. Definiți două variabile ca valori detectate ale senzorului.

```

when Arduino begin
  init lcd I2C address 0x27
  clear lcd
  set lcd back light on
  Declare Global variable Type int Name waterlevel Assigned to 0
  Declare Global variable Type int Name soilhum Assigned to 0
  
```

- Atribuiți cele două valori ale senzorului citit acelor variabile.



- Afișați aceste valori pe LCD.



- Dacă valoarea nivelului apei este mai mică de 700 sau valoarea umidității solului este mai mic de 1200, soneria va alarma.

```

if 700 > variable waterlevel then
  Tone PIN# IO16 frequency NOTE_E5 duration 131
  Tone PIN# IO16 frequency NOTE_A5 duration 131
  Tone PIN# IO16 frequency NOTE_B5 duration 131
  NoTone IO16
endif
if 1200 > variable soilhum then
  Tone PIN# IO16 frequency NOTE_D3 duration 131
  Tone PIN# IO16 frequency NOTE_F3 duration 131
  Tone PIN# IO16 frequency NOTE_A3 duration 131
  NoTone IO16
endif

```

- Când valoarea umidității solului este mai mică de 1200, dar nivelul apei valoarea este mai mare de 700, pompa de apă va iriga automat ferma.

```

if variable soilhum < 1200 and variable waterlevel > 700 then
  relay pin IO25 output HIGH
  wait 0.4 seconds
  relay pin IO25 output LOW
  wait 0.7 seconds
endif

```

Cod complet:

when Arduino begin

init lcd I2C address 0x27

clear lcd

set lcd back light on

Declare Global variable Type int Name waterlevel Assigned to 0

Declare Global variable Type int Name soilhum Assigned to 0

forever

Set waterlevel variable by read the value of water level IO33

Set soilhum variable by read the value of soil moisture IO32

clear lcd

set lcd cursor position x: 0 y: 0

lcd print WaterLevel:

set lcd cursor position x: 0 y: 1

lcd print SoilHum:

set lcd cursor position x: 11 y: 0

lcd print variable waterlevel

set lcd cursor position x: 11 y: 1

lcd print variable soilhum

if 700 > variable waterlevel then

Tone PIN# IO16 frequency NOTE_E5 duration 131

Tone PIN# IO16 frequency NOTE_A5 duration 131

Tone PIN# IO16 frequency NOTE_B5 duration 131

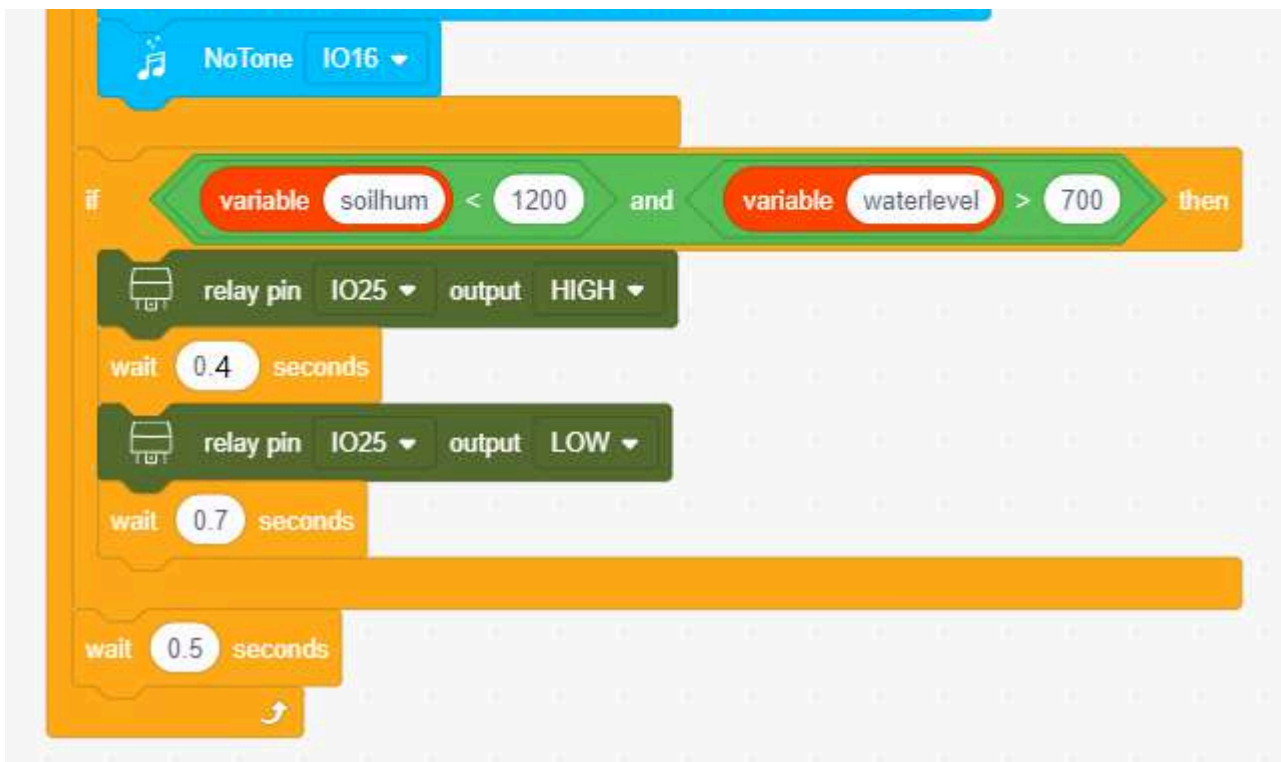
NoTone IO16

if 1200 > variable soilhum then

Tone PIN# IO16 frequency NOTE_D3 duration 131

Tone PIN# IO16 frequency NOTE_F3 duration 131

Tone PIN# IO16 frequency NOTE_A3 duration 131



Rezultatul testului:



- LCD 1602 va afișa valorile curente ale umidității solului și apei nivel. Când umiditatea detectată este mai mică decât pragul setat, implică faptul că solul este arid și începe irigarea automat.
- Când nivelul apei detectat este mai mic decât pragul setat, sistemul de pompare a apei nu funcționează, iar alarmele sonore pentru a anunța că apa este insuficientă.
- Apăsăți butonul pentru a opri alarma.

În concluzie, am realizat un sistem analogic de auto-irigare în acest proiect, care controlează în mod inteligent pornirea și oprirea pompei de apă în funcție de nivelul apei. În aplicație, acest sistem merge de obicei pentru producția casnică și agricolă.

4.10.4 ÎNTREBĂRI FRECVENTE

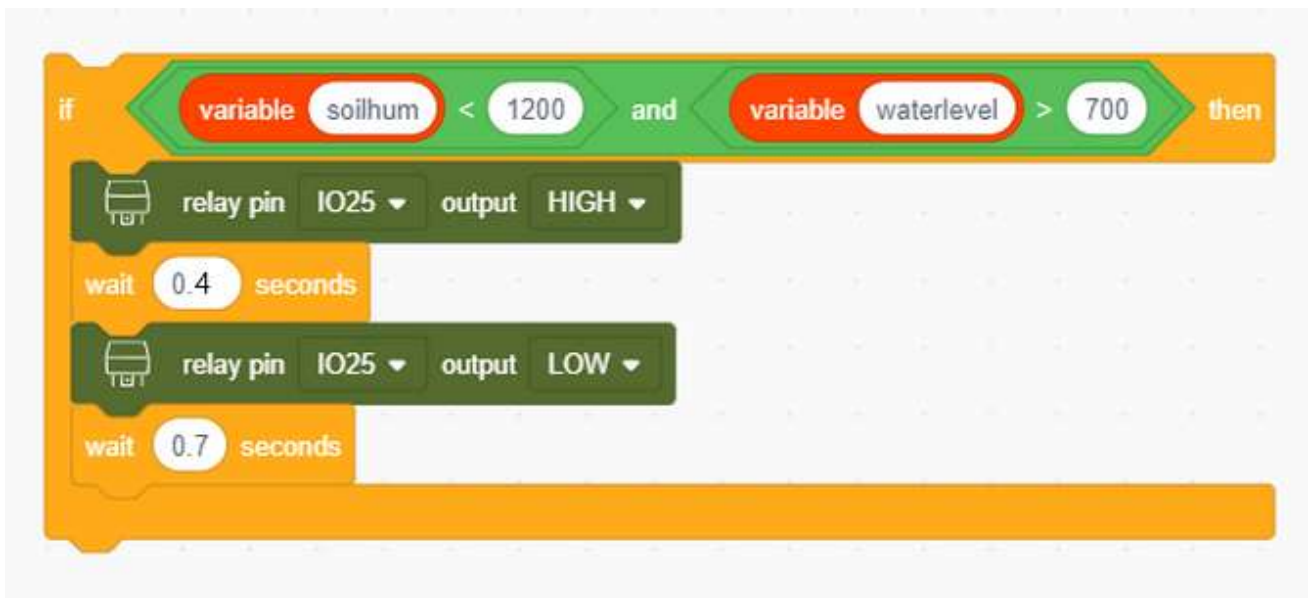
Î: Modulele sunt rezistente la apă?

R: Modulul releului nu este, dar pompa de apă este. Gradul de impermeabilitate a pompei de apă este IP68.

Î: Placa ESP32 este resetată când pompa de apă funcționează.

R: Când pompa de apă funcționează, este necesar mai mult curent decât alte module, prin urmare tensiunea și curentul pot fluctua în circuit. Uneori fluctuațiile pot fi prea mari, rezultând o resetare din cauza tensiunii și curentului în placa de dezvoltare ESP32.

Când utilizați pompa de apă, vă rugăm să urmați exemplul de cod:



Î: Nu reușiți să pompați apă?

R: Sunt necesare mai multe operațiuni de pompare pentru a umple pompa de apă înainte de folosindu-l. Aceste pompe inițiale nu trag de fapt apă, ci pentru a introduceți suficientă apă în pompă. Numai după ce pompa este plină se poate efectua apă. Deci suntem primii pentru umplere, nu pentru pompare.

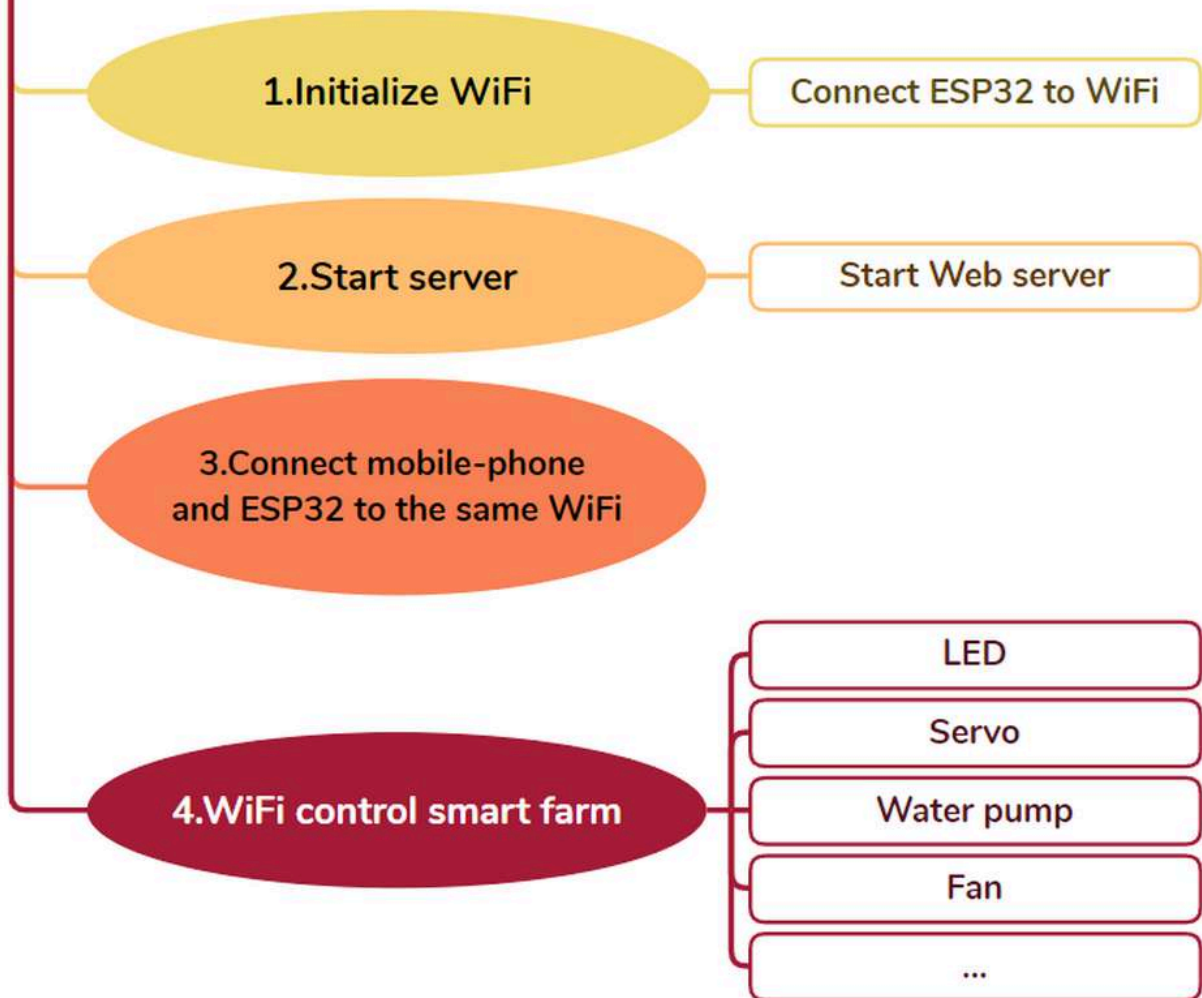
4.11 Proiect: Ferma inteligentă de control WIFI

Fii atent! Nu revărsați apa din bazinele de plastic Experimente. Vărsarea apei pe alți senzori poate provoca un scurtcircuit sau module pentru a rămâne fără loc de muncă. Dacă bateriile se udă, chiar și explozia poate fi întâmplă. Fii foarte atent! Pentru utilizatorii mai tineri, vă rugăm să operați cu părinți. Pentru a garanta securitatea, vă rugăm să respectați îndrumările și siguranța Reglementările.



4.11.1 Diagrama de flux

WiFi Remote Control



4.11.2 Afişarea paginii web WIFI

Descriere:

Placa ESP32 este echipată cu Wi-Fi (2.4G) și Bluetooth (4.2), care permiteți-i să se conecteze cu ușurință la WiFi și să comunice cu alte dispozitive în rețea. În plus, paginile web pot fi afișate în browsere prin ESP32.

ESP32 WiFi



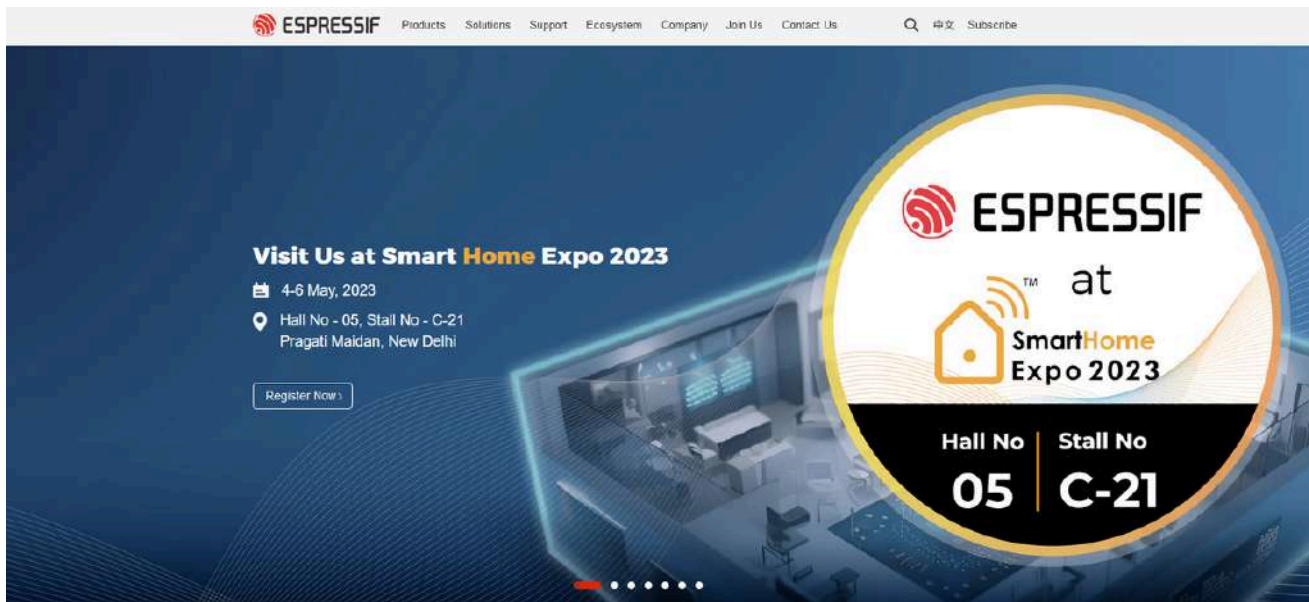
Arduino IDE vă oferă fișierul de bibliotecă <WiFi.h>, care acceptă Configurații Wi-Fi și monitorizare rețea Wi-Fi ESP32.

- **Modul stație de bază** (mod STA sau Wi-Fi pe partea client): În acest mod, ESP32 se conectează la hotspot-ul Wi-Fi (AP).
- **Modul AP** (modul hotspot Soft-AP sau Wi-Fi): În acest mod, altele Dispozitivele Wi-Fi se conectează la ESP32.
- **Modul AP-STA**: În acest mod, ESP32 este un hotspot Wi-Fi, precum și un Dispozitiv Wi-Fi care se conectează la un alt hotspot Wi-Fi.
- Aceste moduri sunt compatibile cu mai multe moduri sigure, cum ar fi WPA, WPA2 și WEP.
- Este capabil să scaneze hotspot-ul Wi-Fi, inclusiv activ și pasiv scanare.
- Acceptă modul promiscuu pentru a monitoriza pachetele Wi-Fi IEEE802.11.



Pentru detalii despre wifi, vă rugăm să consultați:

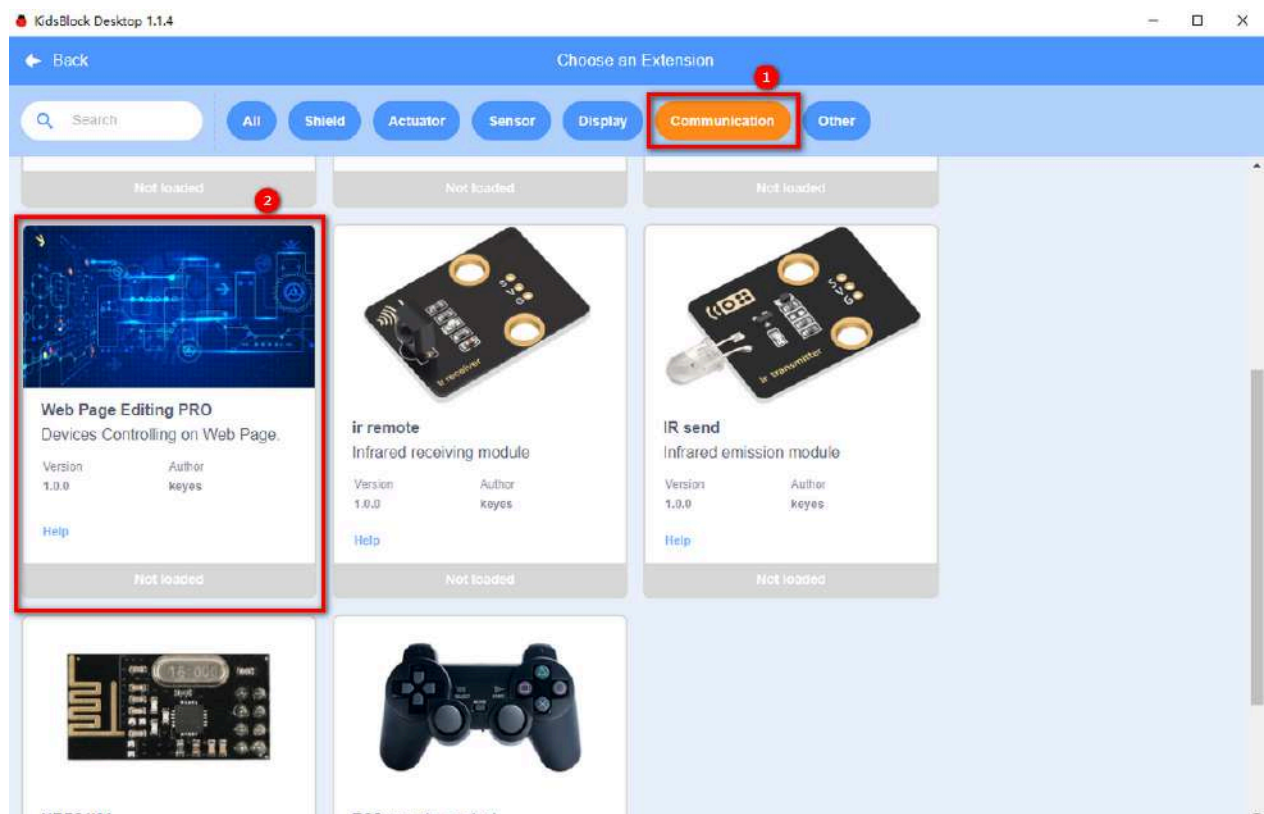
https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/network/esp_wifi.html

Site-ul oficial ESPRESSIF: <https://www.espressif.com.cn/en/home>



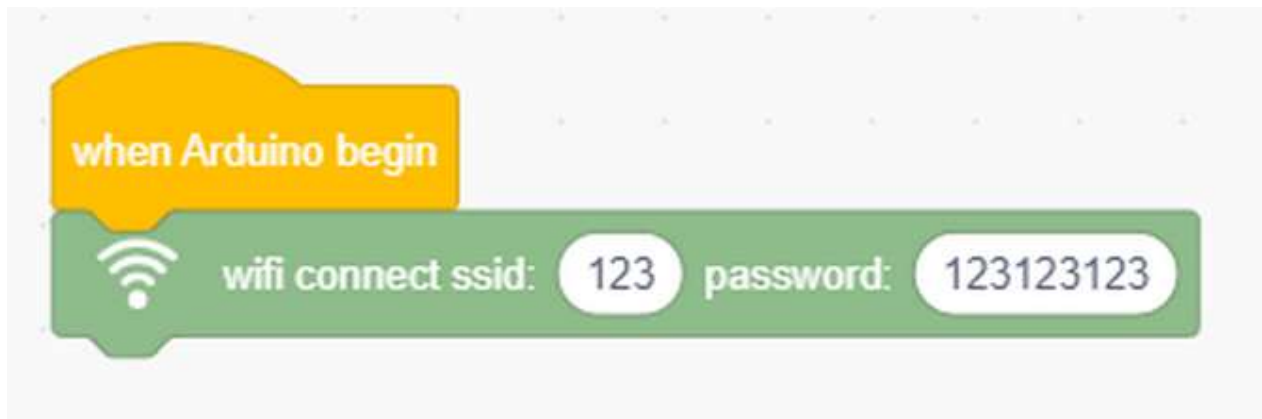
Import bibliotecă

- Clic 
- Faceți clic  pentru a alege "Web Page Editing PRO" și bibliotecile vor fi încărcate.



Cod de testare:

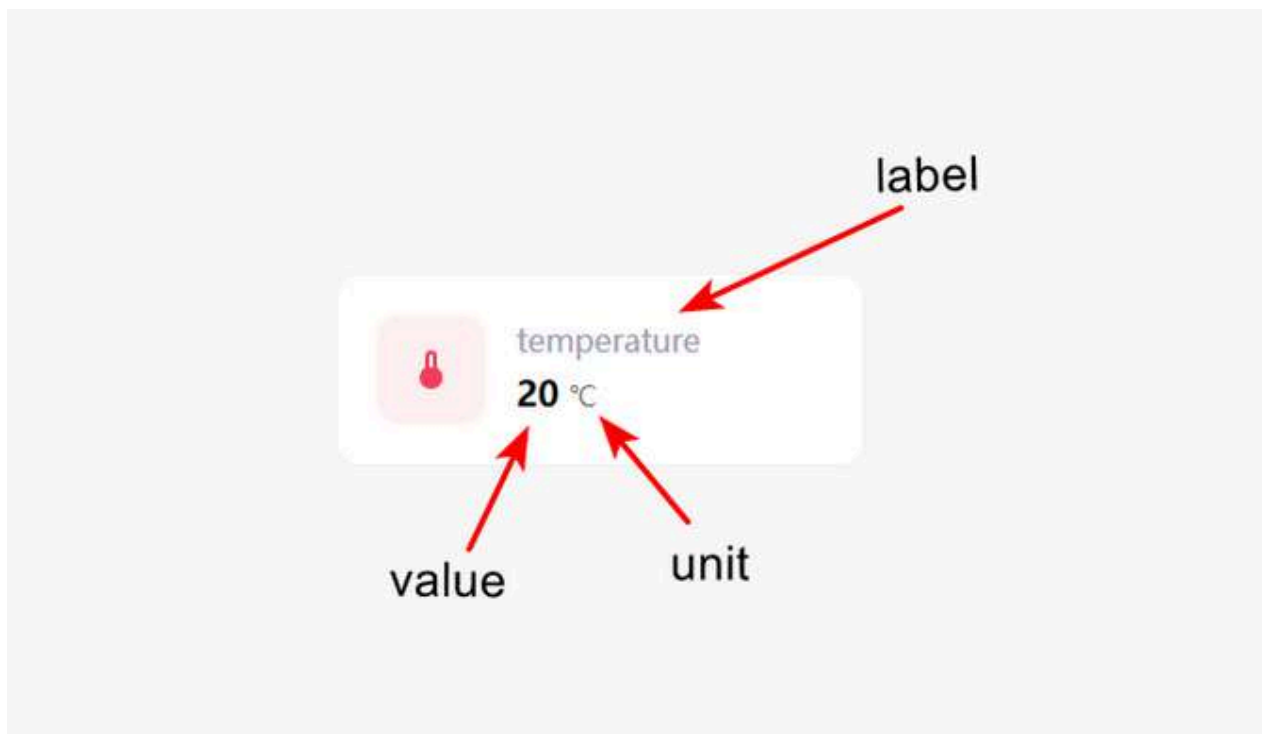
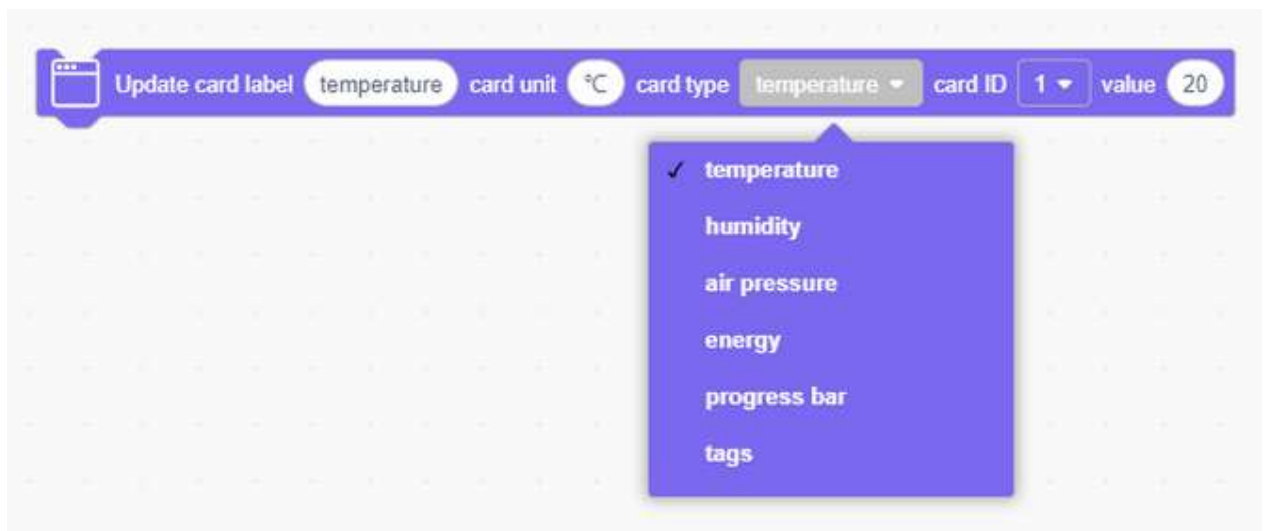
- Conectați-vă la hotspot-ul WiFi, introduceți SSID-ul și parola.



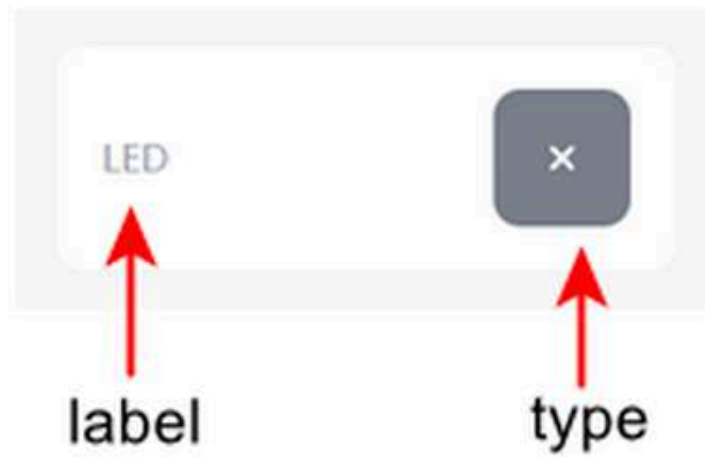
- Afișați adresa IP pe LCD



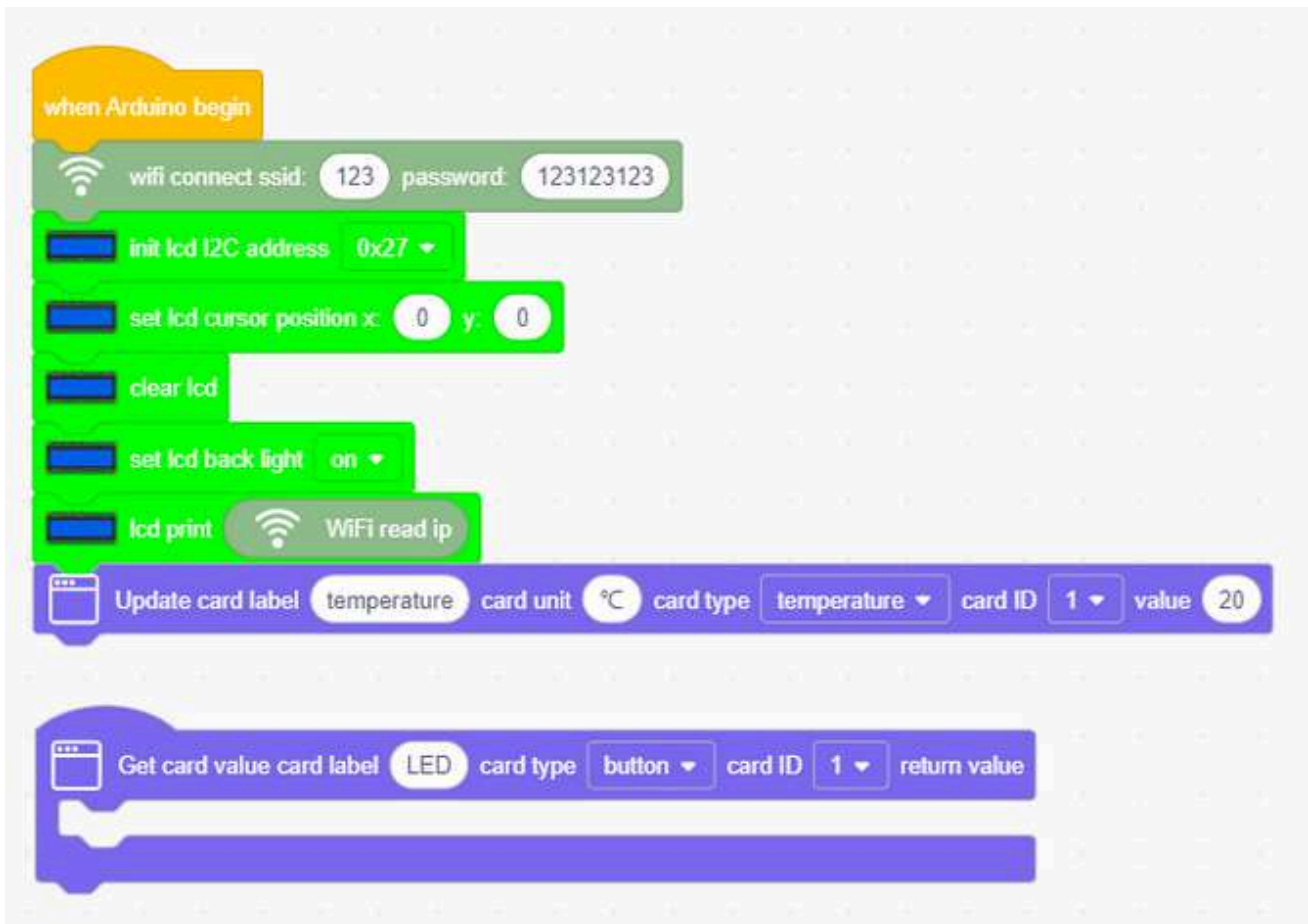
- Definiți componentele unei pagini web numite temperatură (unitate: °C)



- Adăugați un buton numit "buton"



Cod complet:



Vizitați site-ul web

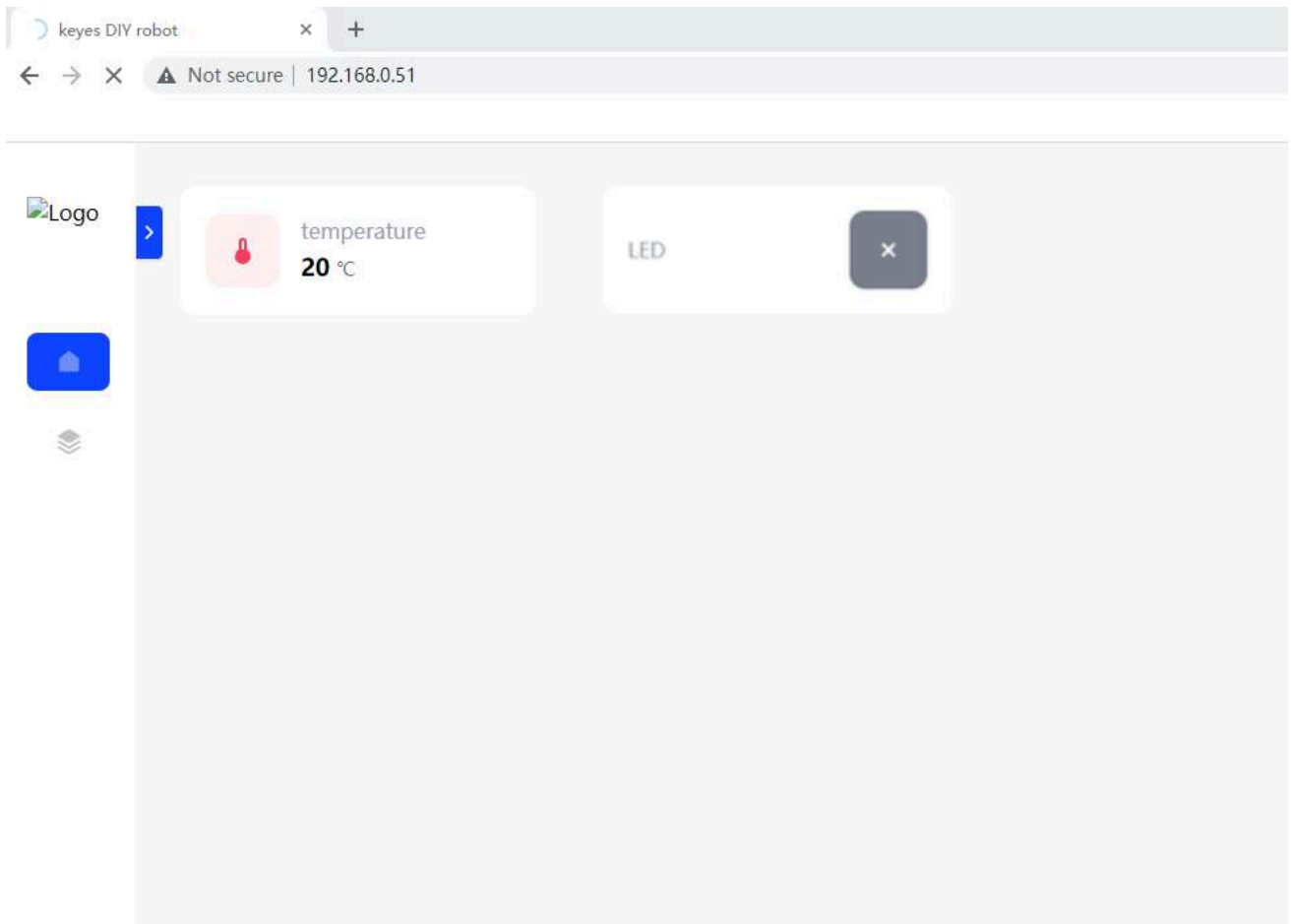
Odată conectat la WiFi, puteți utiliza biblioteca serverului web a ESP32 pentru a servește pagini web. În exemplul următor, vom crea un web simplu pentru a afișa o informație fixă despre temperatură:

Nu în ultimul rând, puteți deschide adresa IP în browser pentru a vizita pagină web. În exemplul nostru de cod, vă rugăm să introduceți "[http://\[Adresa IP a ESP32\]](http://[Adresa IP a ESP32])" pentru a vizita site-ul web.

NOTĂ: Când PC-ul, telefoanele mobile și placa ESP32 sunt conectate la unul rețea, puteți vizita acest site web la PC și telefoane în același timp.

Iată adresa IP ESP32 a dvs.

PC:



Telefon mobil:



temperature

20 °C

LED



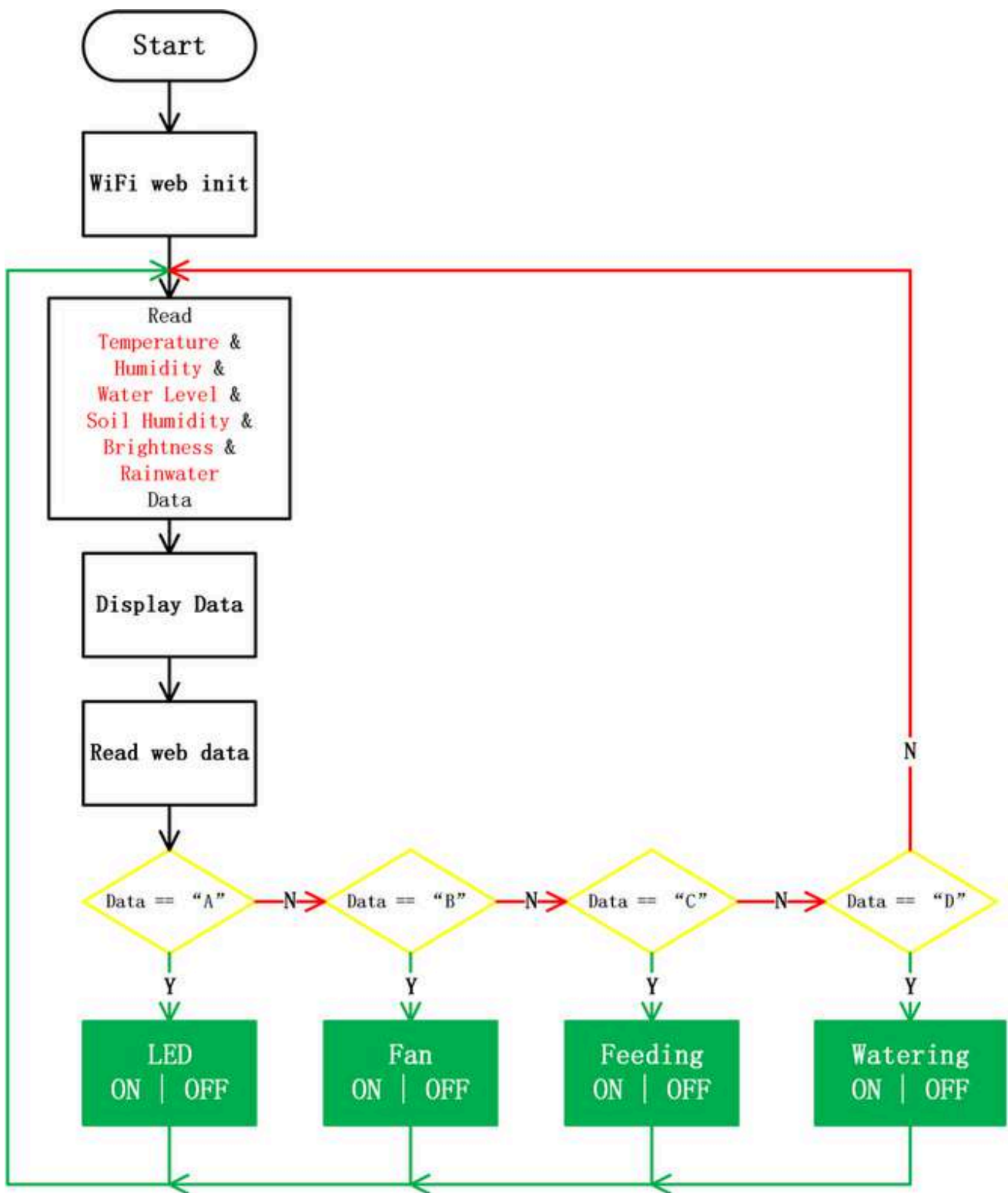


keyes



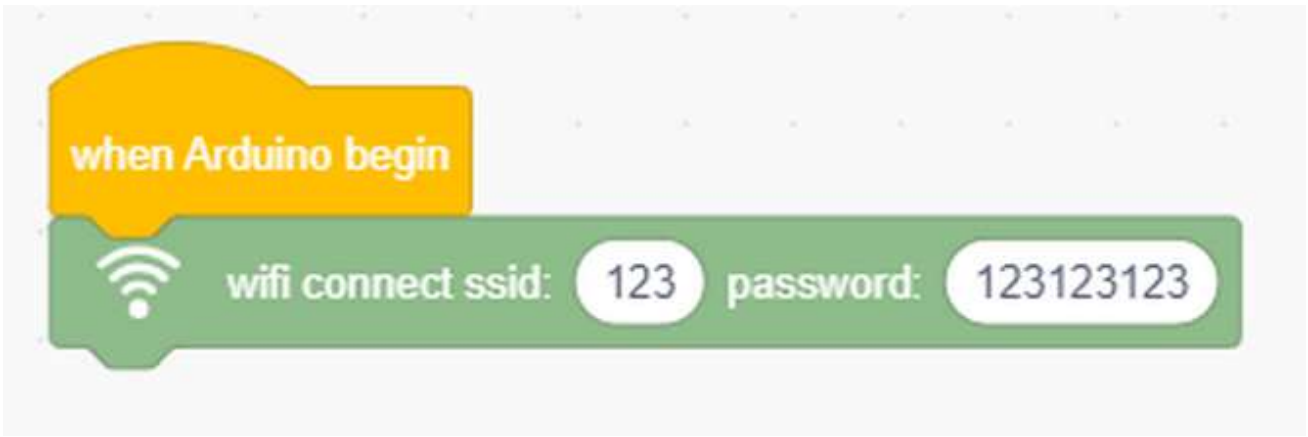
4.11.3 Ferma inteligentă de control WIFI

Flux de cod:



Încărcați codul.

SSID și PAROLA sunt necesare pentru a modifica numele dvs. parolă:



Cod complet:

```

when Arduino begin
  wifi connect ssid: TEST12 password: 88888888
  init lcd I2C address: 0x27
  set lcd cursor position x: 0 y: 0
  clear lcd
  set lcd back light on
  lcd print WiFi read ip
  serial 0 begin baudrate 115200
  set pin IO27 mode output
  init dht11 1 pin IO17 mode dht11
  Declare Global variable Type int Name item1 Assigned to 0
  Declare Global variable Type int Name item2 Assigned to 0
  Declare Global variable Type int Name item3 Assigned to 0
  forever
    Update card label temperature card unit °C card type temperature card ID 1 value dht11 1 read temperature
    Update card label Air Humidity card unit %RH card type humidity card ID 2 value dht11 1 read humidity
    Update card label Soil Humidity card unit card type humidity card ID 3 value read the value of soil moisture IO32
    Update card label Brightness card unit card type tags card ID 4 value read the value of light IO34
    Update card label Water level card unit card type tags card ID 5 value read the value of water level IO33
    Update card label Rainwater card unit card type tags card ID 6 value read the value of drop IO35
    wait 1 seconds
  end forever

  Get card value card label LED card type button card ID 7 return value
  if variable item1 = 0 then
    Set item1 variable by 1
    LED pin IO27 output HIGH
    serial 0 print LED HIGH warp
  else
    Set item1 variable by 0
    LED pin IO27 output LOW
    serial 0 print led low warp

  Get card value card label Watering card type button card ID 8 return value
  relay pin IO25 output HIGH
  wait 0.4 seconds
  relay pin IO25 output LOW
  wait 0.7 seconds

  Get card value card label Feeding card type button card ID 10 return value
  if variable item3 = 0 then
    Set item3 variable by 1
    servo PIN# IO26 channel CH1 (LT0) degree 80 delay 200
  else
    Set item3 variable by 0
    servo PIN# IO26 channel CH1 (LT0) degree 180 delay 200

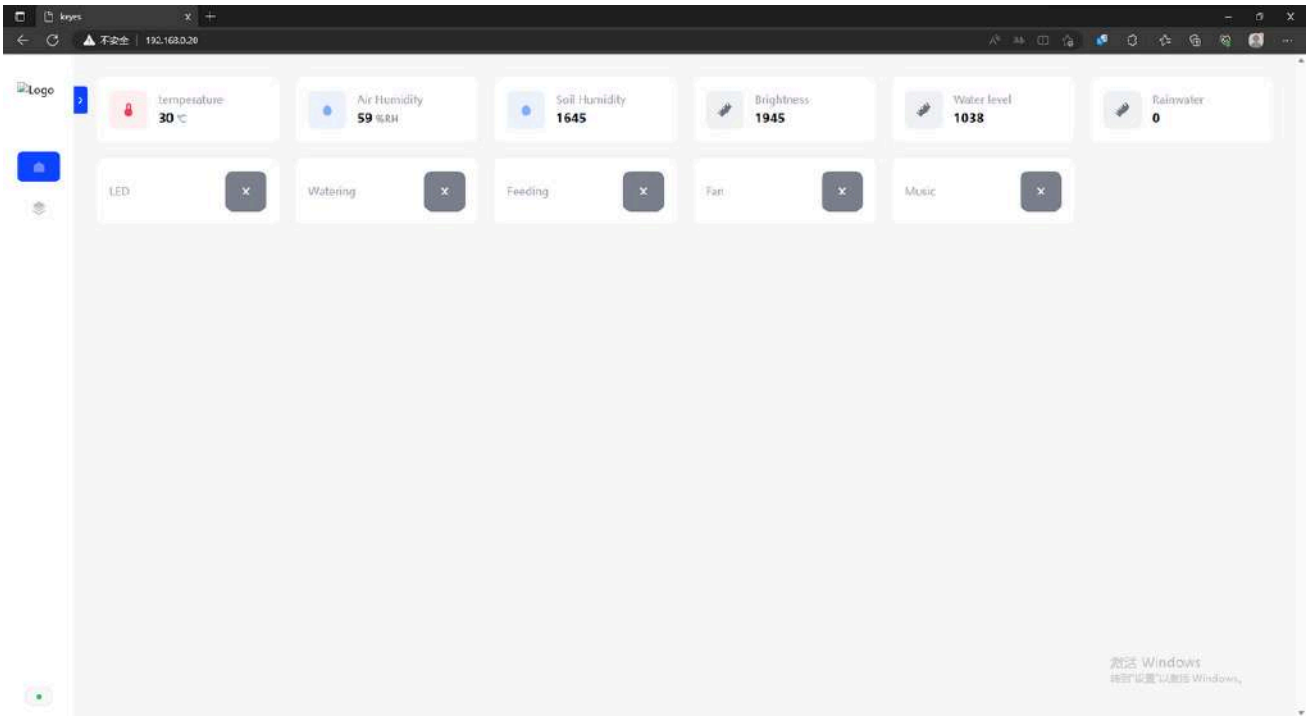
  Get card value card label Fan card type button card ID 9 return value
  if variable item2 = 0 then
    Set item2 variable by 1
    fan INA# IO18 State LOW INB# IO19 analogWrite 70
  else
    Set item2 variable by 0
    fan INA# IO18 State LOW INB# IO19 analogWrite 0

  Get card value card label Music card type button card ID 11 return value
  Tone PIN# IO16 frequency NOTE_C3 duration 131
  wait 0.3 seconds
  NoTone IO16
  wait 0.3 seconds

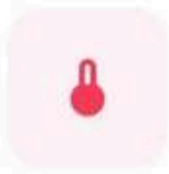
```

Rezultat:

PC:



Telefon mobil:



temperature
30 °C



Air Humidity
60 %RH



Soil Humidity
1765



Brightness
1962



Water level
1043



Rainwater



Watering

0

LED



Watering



Feeding



Fan



Music



senzorului și controlează LED-ul și ventilatorul.

Valorile senzorului	Dispozitive controlabile
Temperatura (°C)	LED
Umiditate (%RH)	Evantai
Nivelul apei (%)	Cutie de hrănire
Precipitații (%)	Pompa de apă
Luminozitate (0~4095)	
Umiditatea solului (%)	

Cu placa de dezvoltare ESP32, am învățat cum să creăm un web pentru a afișa valorile senzorului, cum ar fi temperatura, umiditatea, apa nivelul și umiditatea solului și putem controla, de asemenea, luminile LED, ventilatoarele, cutii și pompe de alimentare. Mai mult, aceste operațiuni pot fi de la distanță finisat prin telefoane mobile sau computere.



În acest proiect, stimulăm o fermă inteligentă cu gestiune. O astfel de tehnologie facilitează controlul echipamentelor și îmbunătățește eficiența și calitatea agriculturii, ceea ce face ca internetul Lucruri, informatizare, automatizare și inteligență posibilă.

4.11.4 ÎNTREBĂRI FRECVENTE

Î: Wi-Fi nu reușește să fie conectat întotdeauna.

R: Mutați ESP32 în partea laterală a routerului și reporniți placa și doar Așteptați răbdare să așteptați. Dacă tot nu se conectează, vă rugăm să verificați dacă numele și parola WiFi sunt corecte.

Î: Răspunsul este lent în timpul opterărilor de la distanță pe pagina web.

R: Motive posibile:

- Resursele CPU ale routerului sunt insuficiente din cauza mai multor Conexiuni. Vă rugăm să reporniți routerul pentru a încerca o reconectare.
- Routerul funcționează mult timp. Vă rugăm să reporniți routerul.
- Interferențe wireless. Semnalul wireless este instabil, așa că vă rugăm să nu folosiți-l prin perete.

Pentru cunoștințe despre routere, vă rugăm să căutați pe Google singur.

Î: Nu reușiți să pompați apă?

R: Sunt necesare mai multe operațiuni de pompare pentru a umple pompa de apă înainte de folosindu-l. Aceste pompe inițiale nu trag de fapt apa, ci pentru a introduceți suficientă apă în pompă. Numai după ce pompa este plină se poate efectua apă. Deci suntem primii pentru umplere, nu pentru pompare.

4.12 Proiect: Ferma inteligentă de control APP

Fii atent! Nu revărsați apa din bazinele de plastic Experimente. Vărsarea apei pe alți senzori poate provoca un scurtcircuit sau module pentru a rămâne fără loc de muncă. Dacă bateriile se udă, chiar și explozia poate fi întâmplă. Fii foarte atent! Pentru utilizatorii mai tineri, vă rugăm să operați cu părinți. Pentru a garanta securitatea, vă rugăm să respectați îndrumările și siguranța Reglementările.



4.12.1 Descriere

Sistemul de management APP este capabil să monitorizeze mai multe indici în timp real de ferma, cum ar fi temperatura și umiditatea, nivelul apei din piscină, solul umiditate, intensitate luminoasă și precipitații.

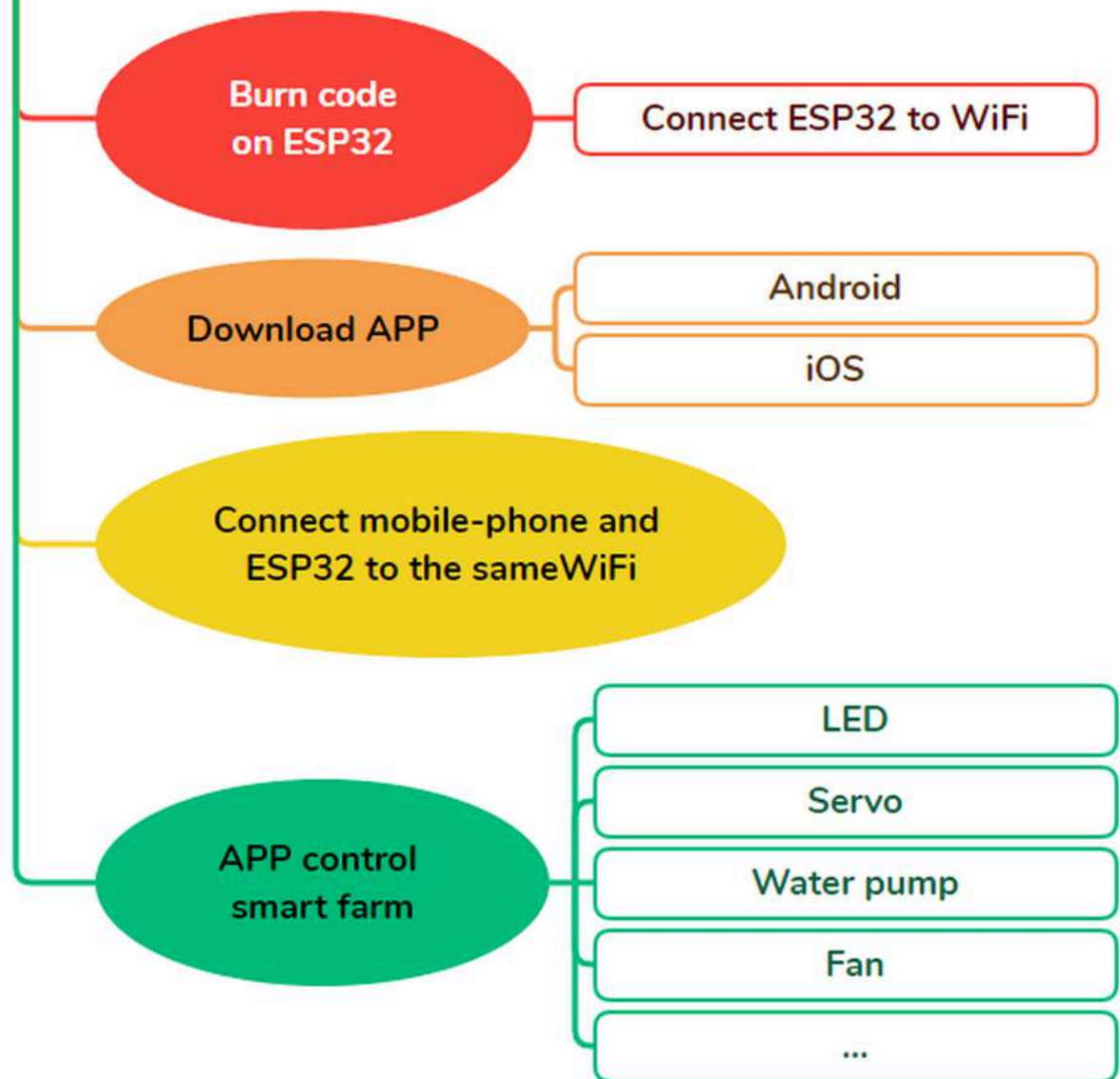
Între timp, controlează și LED-urile pentru iluminat, pompa de apă pentru irigare, Cutie de alimentare pentru alimentare și ventilator pentru reglarea temperaturii și umidității.



Aceste funcții pot fi realizate printr-o aplicație de pe telefon, facilitând managementul fermei. Pentru mai multă inteligență, un buzzer a fost adoptat și ca alarma.

4.12.2 Diagrama de flux

APP Remote Control



4.12.3 Codul de testare

Flux de cod:

Start

ESP32 Connected to WiFi

Read
Temperature & Humidity &
Water Level & Soil
Humidity &
Brightness & Rainwater
Data

Display data to APP

Read data from the APP

Data == "a"

Data == "A"

LED OFF

LED ON

Data == "c"

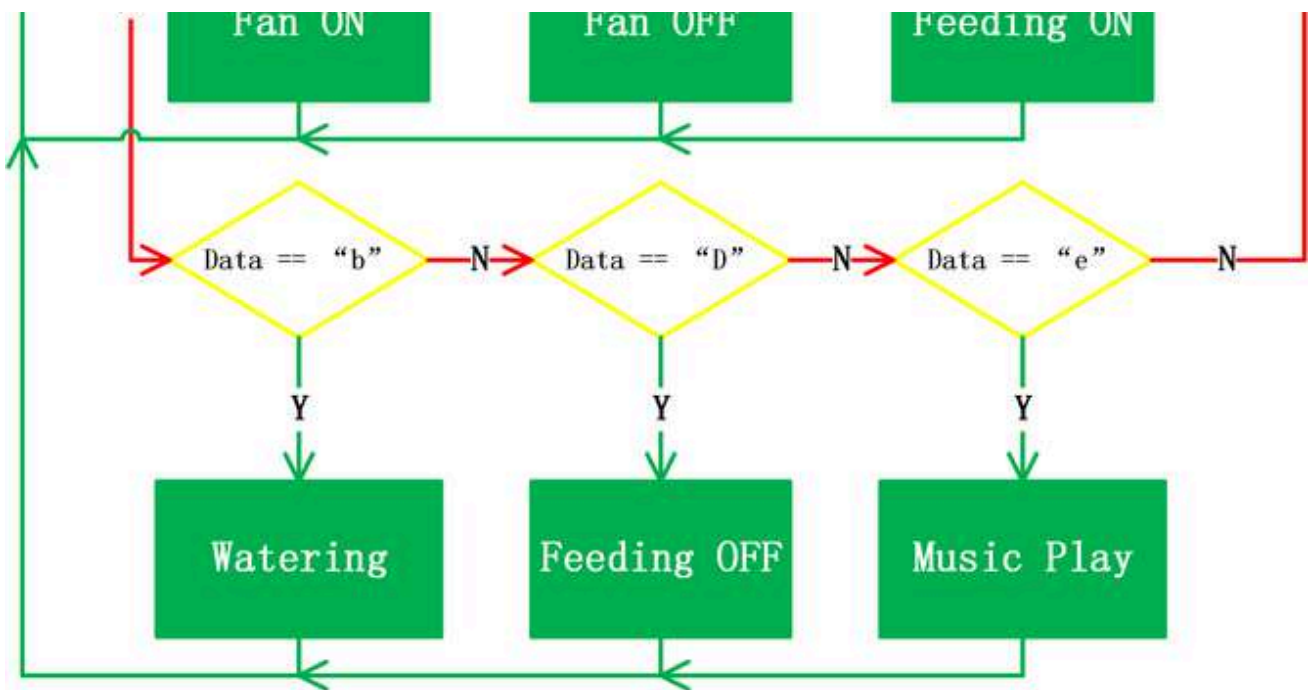
Data == "C"

Data == "d"

LED ON

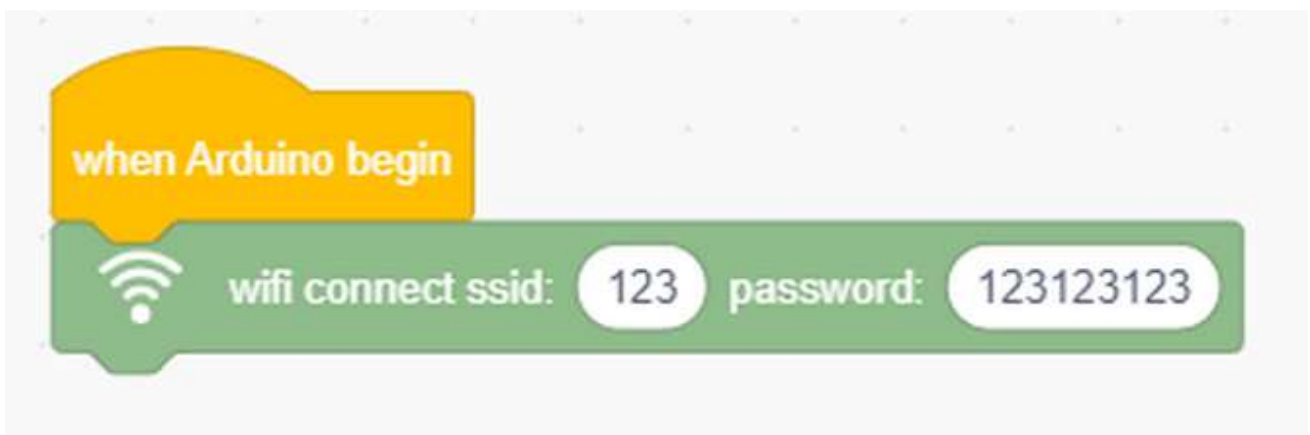
LED OFF

LED ON



Ardere cod pe ESP32:

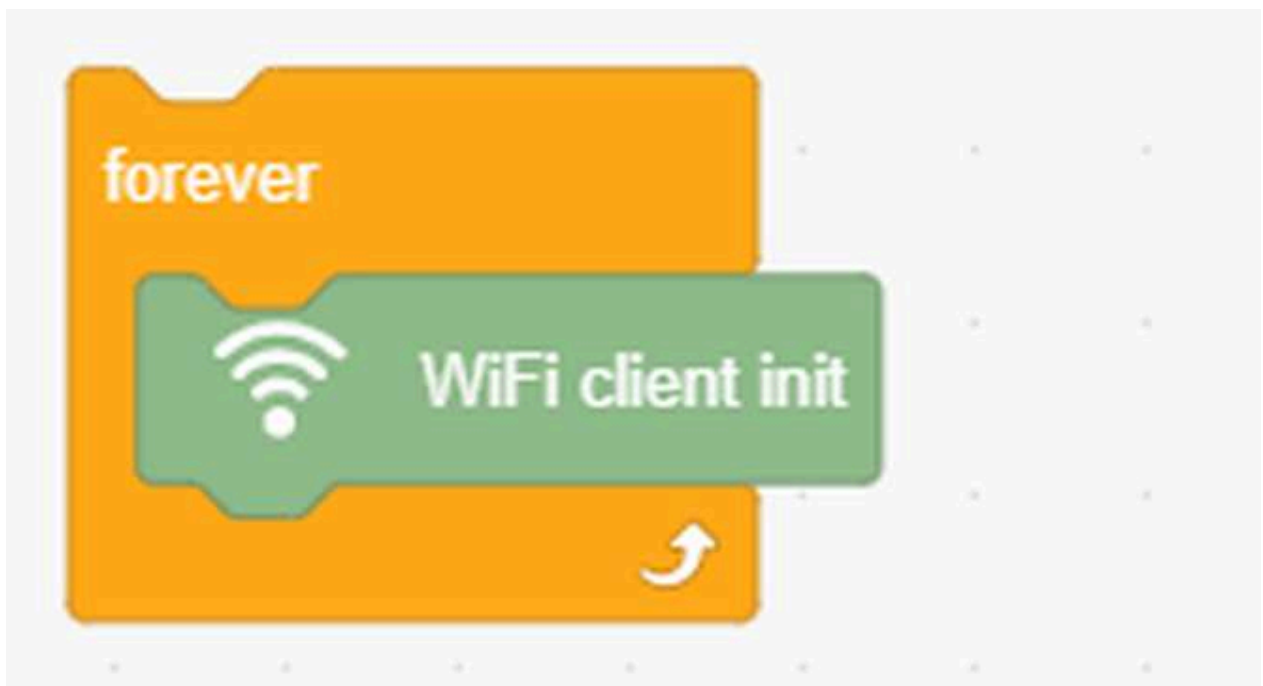
- Conectați ESP32 la WiFi. În următorul cod, **ssid** și **pwd** sunt, respectiv, numele și parola WiFi



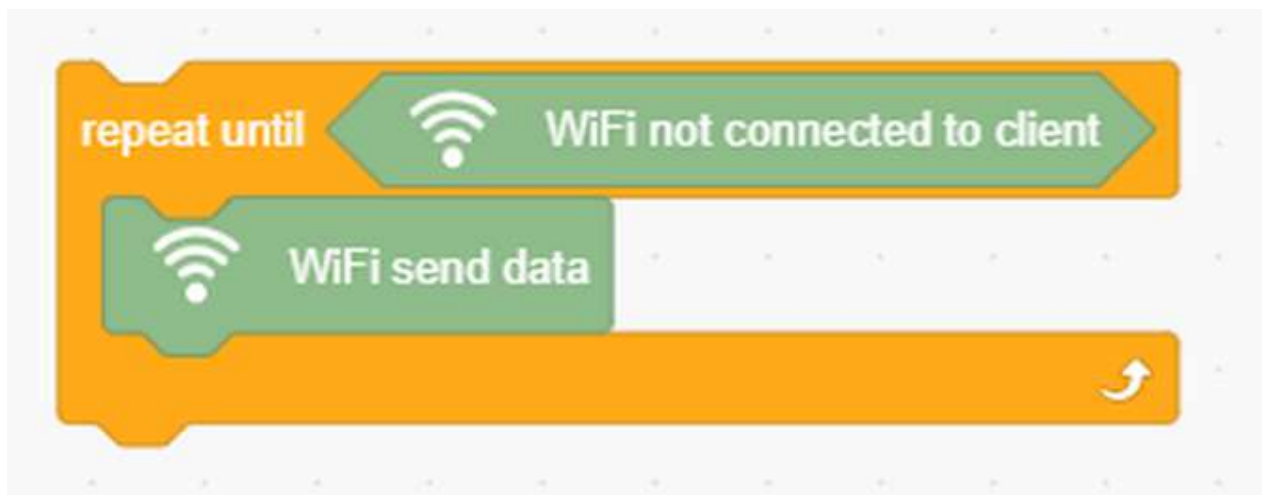
- LCD afișează adresa IP



- Initialize wifi server. After initialization, ESP32 and APP can communicate with each other through WIFI.



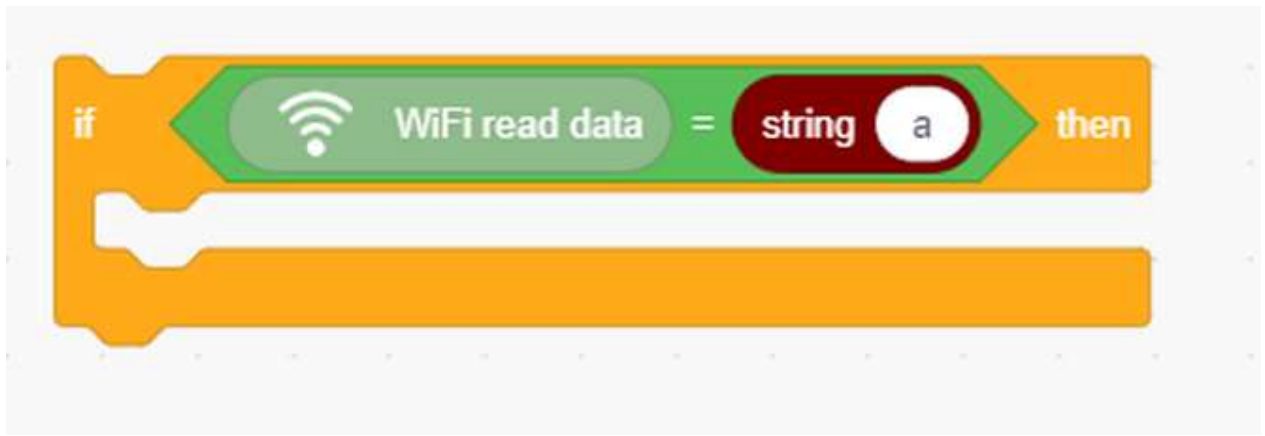
- Verificați dacă Wi-Fi este conectat la client/APP



- Trimiteți date în timp real ale senzorilor către APP:



- ESP32 primește date de la APP și le determină. NOTĂ: Toate datele sunt în formatul String.



Cod complet:

when Arduino begin

serial 0 begin baudrate 115200

wifi connect ssid: 123 password: 88888888

init lcd I2C address 0x27

clear lcd

set lcd back light on

set lcd cursor position x: 0 y: 0

lcd print WiFi read ip

forever

WiFi client init

repeat until WiFi not connected to client

WiFi send data

if WiFi read data = string a then

LED pin IO27 output HIGH

if WiFi read data = string A then

LED pin IO27 output LOW

if WiFi read data = string b then

relay pin IO25 output HIGH

wait 0.4 seconds

relay pin IO25 output LOW

wait 0.7 seconds

if WiFi read data = string c then

wait 0.8 seconds

fan INA# IO18 State LOW INB# IO19 analogWrite

wait 0.2 seconds

```
if WiFi read data = string C then
  fan INA# IO18 State LOW INB# IO19 analogWrite 0
if WiFi read data = string d then
  servo PIN# IO26 channel CH1 (LT0) degree 80 delay 200
if WiFi read data = string D then
  servo PIN# IO26 channel CH1 (LT0) degree 180 delay 200
if WiFi read data = string e then
  Tone PIN# IO16 play music Birthday
  NoTone IO16
```

4.12.4 APLICAȚIE

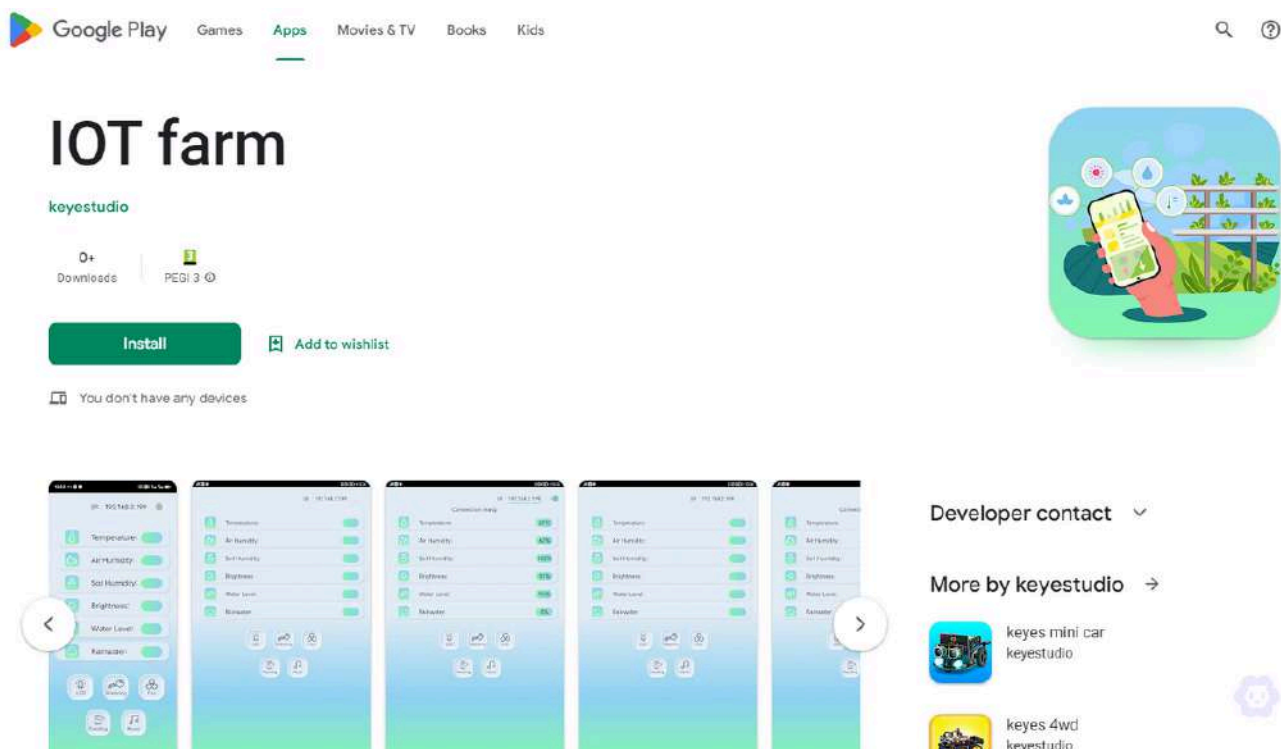
Descărcarea aplicației:



IOT farm

Android:

- Deschideți Google Play și căutați ferma IOT pentru a descărca.



- În fișierele furnizate, este inclus pachetul de instalare Android apk:



IOS:

Căutați ferma IOT în APP Store și atingeți pentru a descărca.

Interfața aplicației

IP: 192.168.3.7



Temperature:



Air Humidity:



Soil Humidity:



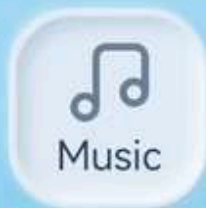
Brightness:



Water Level:



Rainwater:



Descrierea funcției APP:

1. Când telefonul și placa ESP32 se conectează la același WIFI, trebuie să introduceți adresa IP în partea dreaptă sus pentru a le conecta.



IP: 192.168.3.2

2. Afășează valoarea temperaturii fermei în timp real.



Temperature:



3. Afășează valoarea umidității fermei în timp real.



Air Humidity:



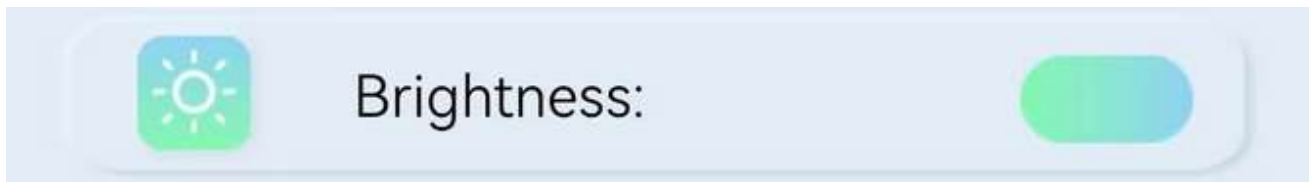
4. Afășează valoarea umidității solului fermei în timp real.



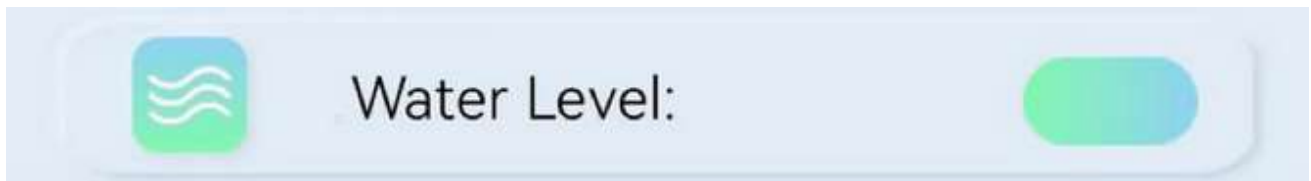
Soil Humidity:



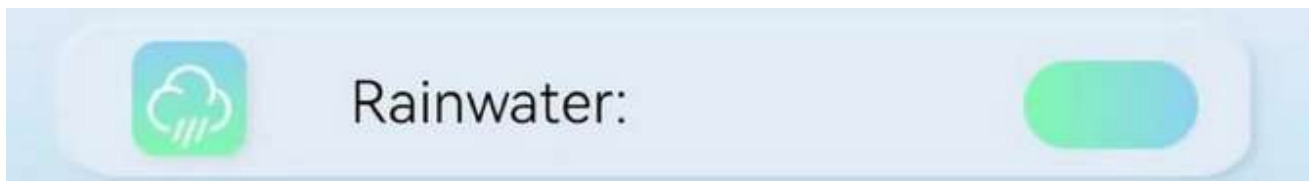
5. Afășează valoarea luminozității soarelui a fermei în timp real.



6. Afîșează nivelul apei fermei în timp real.



7. Afîșează valoarea analogică a precipitațiilor fermei în timp real.



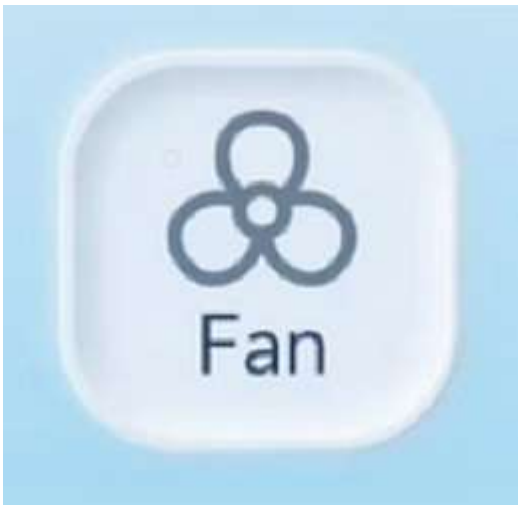
8. LED de control.



9. Controlul irigații prin pompă de apă.



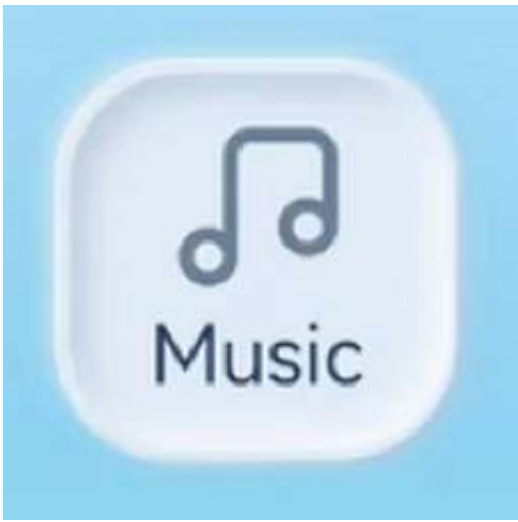
10. Controlați ventilatorul pentru a regla temperatura.



11. Servo de control pentru a deschide sau închide cutia de alimentare.



12. Controlați soneria pentru a reda muzică.



4.12.5 ÎNTREBĂRI FRECVENTE

Î: Wi-Fi nu reușește să fie conectat întotdeauna.

R: Mutați ESP32 în partea laterală a routerului și reporniți placa și doar Aveți răbdare să așteptați. Dacă tot nu se conectează, vă rugăm să verificați dacă numele și parola WiFi sunt corecte.

Î: APP nu reușește să se conecteze la ESP32.

R: Vă rugăm să vă asigurați că APP și ESP32 sunt conectate la același WiFi.

Î: Nu reușiți să pompați apă?

R: Sunt necesare mai multe operațiuni de pompare pentru a umple pompa de apă înainte de folosindu-l. Aceste pompe inițiale nu trag de fapt apa, ci pentru a introduceți suficientă apă în pompă. Numai după ce pompa este plină se poate efectua apă. Deci suntem primii pentru umplere, nu pentru pompare.

5. ÎNTREBĂRI FRECVENTE

Î: Cu ce tip de baterii ar trebui să fie echipat acest kit?

R: Șase baterii AA, fiecare cu diametrul de 14 mm și înălțimea 49MM. Vă rugăm să instalați bateriile într-un mod corect și să nu le inversați! Pentru cursanții mai mici, vă rugăm să operați sub însoțirea părinților.

Î: Apare o eroare la inscripționarea programelor pe placa de bază ESP32.

Un:

- Vă rugăm să verificați dacă portul COM este corect.
- Vă rugăm să verificați dacă placa selectată este corectă.

Î: Acest kit se poate extinde la alte module?

R: Da. Când extindeți la alte module, vă rugăm să verificați descrierea pinului la asigurați-vă că pini ESP32 funcționează normal.

Î: Apare o eroare la importul bibliotecii <Wire.h>.

R: Când instalați placa de dezvoltare ESP32 pe arduino IDE, <Wire.h> Biblioteca va fi importată automat, deci nu trebuie să o adăugați repetat.