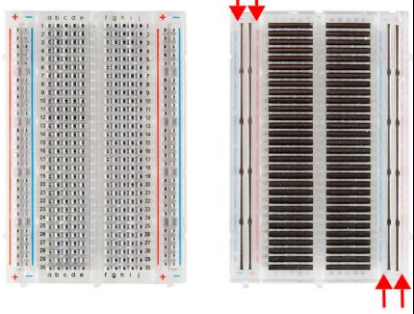
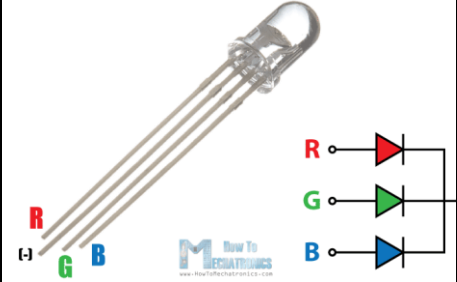
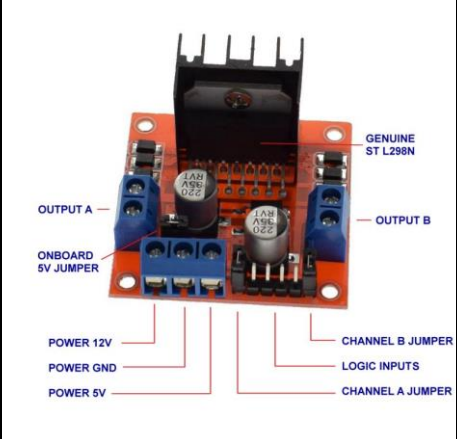


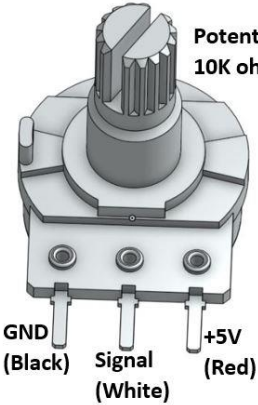
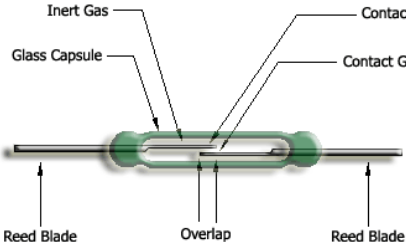
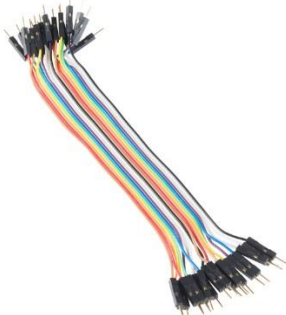
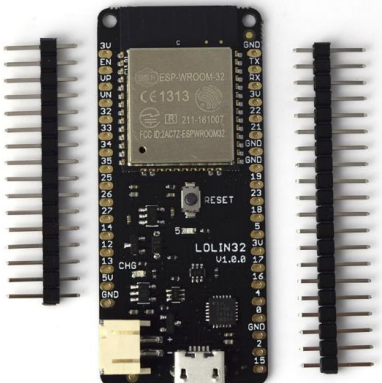


# Miele Washing Machine Experience Kit

## What's in the kit

Picture	Description
	<p><b>Breadboard</b>(solderless breadboard) este o placa pe care se pot pune elemente de circuit. Aceasta placa se folosește în fazele incipiente ale proiectelor, pentru teste. Pe aceasta nu se lipește nimic ceea ce permite modificarea facilă a circuitului sau remedierea erorilor. În imaginea alăturată, este un breadboard: pentru o alimentare ușoară de obicei se alimentează cele 2 perechi de rânduri + și -. Se poate observa o linie care împarte toate coloanele în două părți. Cele cinci puncte (poziționate simetric față de linia mediană) care formează coloana sunt conectate între ele; astfel <math>A1=B1=C1=D1=E1</math>, dar sunt diferite de <math>F1=G1=H1=I1=J1</math>. Punctele de pe linii de tensiune sunt conectate toate conform liniilor albastre și roșii.</p>
	<p><b>RGB – LED</b> - Un LED (din engleză light-emitting diode, însemnând diodă emițătoare de lumină) este o diodă semiconductoră ce emite lumină la polarizarea directă a joncțiunii p-n. Efectul este o formă de electroluminescență.</p>
	<p>O <b>punte H</b> este un circuit construit din 4 tranzistoare care acționează ca niște întrerupătoare, iar poziția acestora determină sensul de rotație al motorului.</p>
	<p><b>Ecranul LCD</b> (ecran cu cristale lichide) este un modul electronic de afișare ce este folosit într-o gamă largă de aplicații.</p>
	<p>Un <b>buzzer</b> sau un beeper este un dispozitiv de semnalizare audio, care poate fi mecanic, electromecanic sau piezoelectric (piezo pe scurt). Utilizările tipice ale buzzerelor și beeperelor includ dispozitive de alarmă, cronometre și confirmarea introducerii de către utilizatori, cum ar fi un clic de mouse sau o apăsare de tastă.</p>

 <p><b>Potentiometer</b> 10K ohms, Linear</p> <p>GND (Black)    Signal (White)    +5V (Red)</p>	<p>Un <b>potențiomtru</b> este un instrument pentru variația potențialului electric (tensiune) într-un circuit. Înainte de introducerea bobinei mobile și a voltmetrelor digitale, potențiometrele (d) au fost utilizate la măsurarea tensiunii electrice.</p>
 <p>Inert Gas    Contact Glass Capsule    Contact Gap Reed Blade    Overlap    Reed Blade</p>	<p>Senzorul de prezență(<b>Reed Switch</b>) este un comutator electric acționat de un câmp magnetic aplicat. Se compune dintr-o pereche de contacte pe stâlpi de metal feromagnetic într-un plic de sticlă închis ermetic. Contactele pot fi deschise în mod normal și închise când este prezent un câmp magnetic.</p>
	<p><b>Jumper Wires</b> (Breadboard Wires) este un cablu electric sau un grup de cabluri este folosit pentru a lega componentele de pe breadboard fără a fi necesară lipirea acestora.</p>
	<p><b>ESP32</b> este o serie de sisteme low-cost, de joasă putere pe un microcontroler cu chip cu Wi-Fi integrat și Bluetooth cu dublă mod.</p> <p>Specs here:  <a href="https://www.espressif.com/sites/default/files/documentati on/esp32-wroom-32_datasheet_en.pdf">https://www.espressif.com/sites/default/files/documentati on/esp32-wroom-32_datasheet_en.pdf</a></p>

# Overview of Arduino Integrated Development Environment

Programele Arduino pot fi scrise în orice limbaj de programare cu un compilator capabil să producă un cod mașină binar. Atmel oferă un mediu de dezvoltare pentru microcontrolerele sale, AVR Studio și mai nou, Atmel Studio.

Proiectul Arduino oferă un mediu integrat de dezvoltare (IDE), care este o aplicație cross-platform, scrisă în Java. Acesta își are originile în mediul de dezvoltare pentru limbajul de programare Processing și în proiectul Wiring.

Este proiectat pentru a introduce programarea în lumea artiștilor și a celor nefamiliarizați cu dezvoltarea software. Include un editor de cod cu funcții ca evidențierea sintaxelor, potrivirea acoladelor și spațierea automată și oferă mecanisme simple cu un singur click, pentru a compila și a încărca programele în plăcuța Arduino. Un program scris în IDE pentru Arduino se numește sketch.

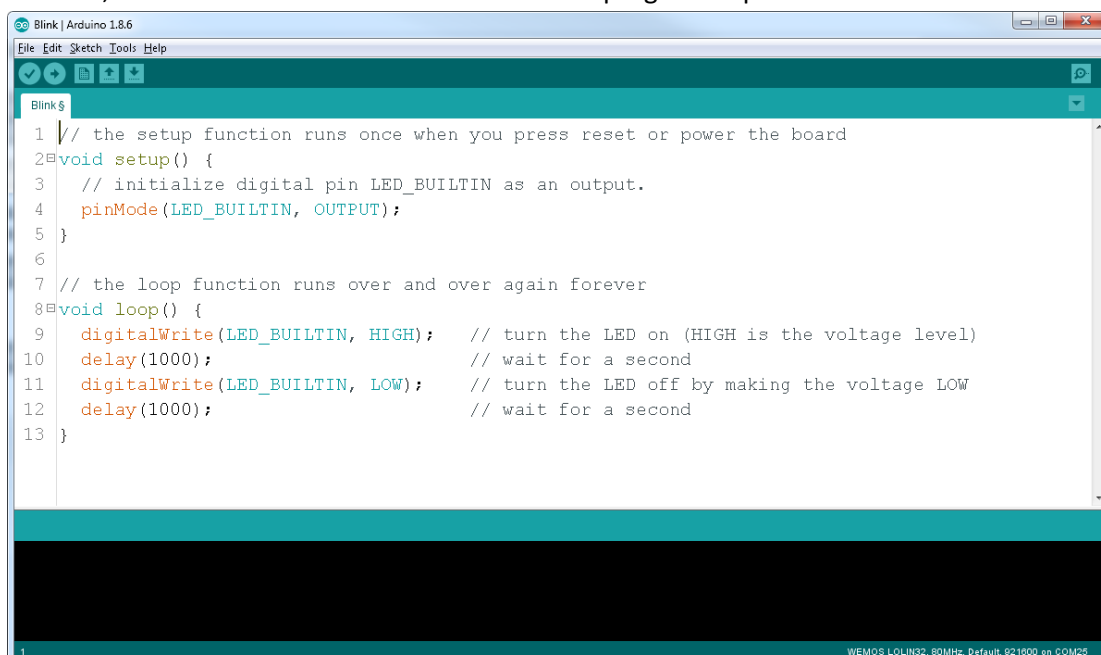
Arduino IDE suportă limbajele de programare C și C++ folosind reguli speciale de organizare a codului. Arduino IDE oferă o librărie software numită Wiring, din proiectul Wiring, care oferă multe proceduri comune de intrare și ieșire. Un sketch tipic Arduino scris în C/C++ este compus din două funcții:

**setup():** o funcție care este rulată o singură dată la începutul programului, când se inițializează setările.

**loop():** o funcție apelată în mod repetat până la oprirea alimentării cu energie a plăcuței.

## Exemplu de program

Un program Arduino tipic pentru un programator începător face ca un LED să se aprindă intermitent. Acest program este încărcat pe placă, în mod normal, de către producător. În mediul de dezvoltare Arduino, utilizatorul ar trebui să scrie un astfel de program după cum urmează:

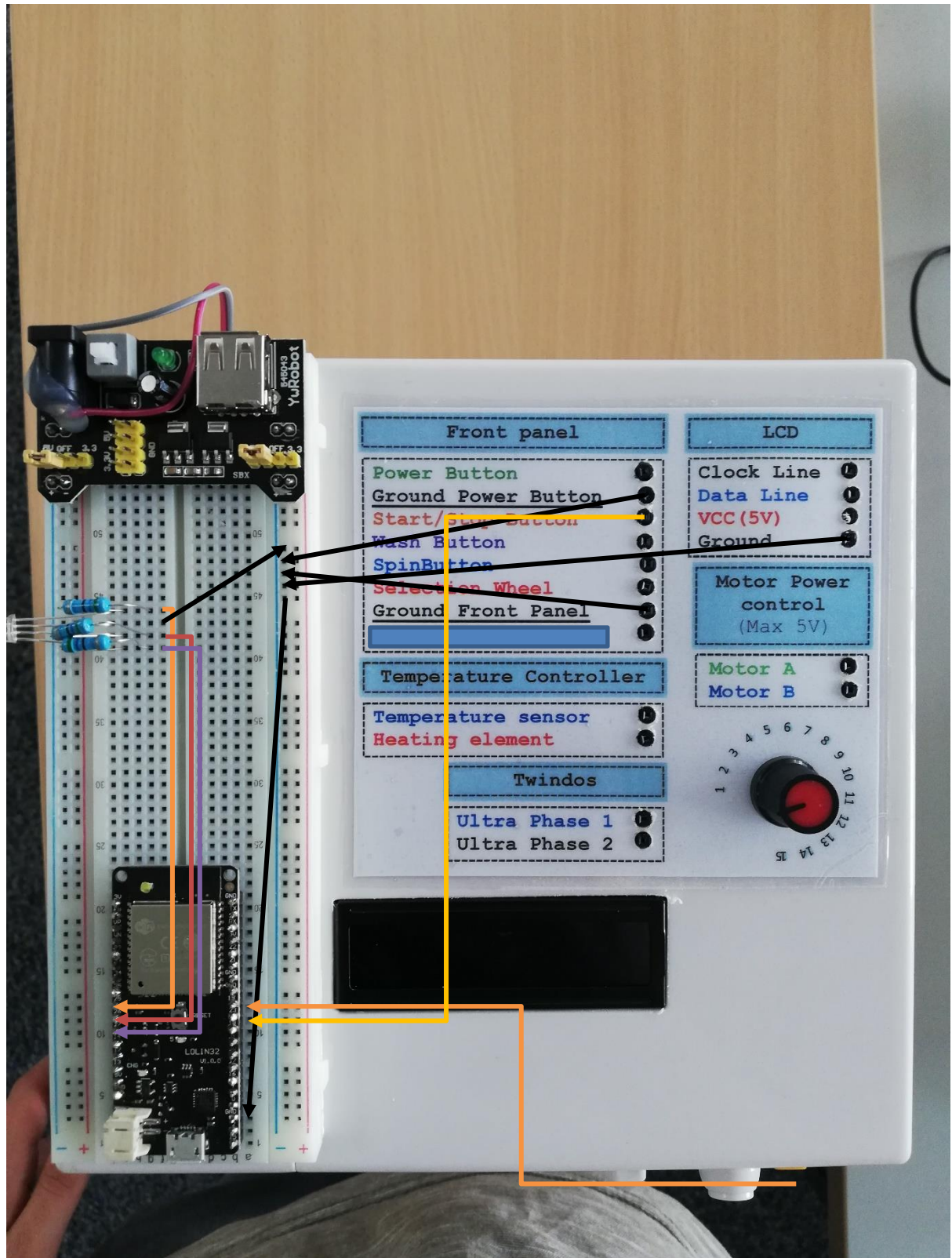
The image is a screenshot of the Arduino IDE interface. The title bar reads "Blink | Arduino 1.8.6". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". Below the menu bar is a toolbar with icons for opening files, saving, running, and other IDE functions. The main text area contains the following C++ code for the "Blink" sketch:

```
1 // the setup function runs once when you press reset or power the board
2 void setup() {
3   // initialize digital pin LED_BUILTIN as an output.
4   pinMode(LED_BUILTIN, OUTPUT);
5 }
6
7 // the loop function runs over and over again forever
8 void loop() {
9   digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
10  delay(1000); // wait for a second
11  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
12  delay(1000); // wait for a second
13 }
```

The bottom status bar indicates "WEMOS LOLIN32, 80MHz, Default, 921600 on COM25".

## Experiment 1 - Funcția de detectare a ușii închise și de pornire la apăsarea unui buton

- 1) De ce ai nevoie: Breadboard, mașină de spălat și fire.
- 2) Setup-ul arată cam așa:

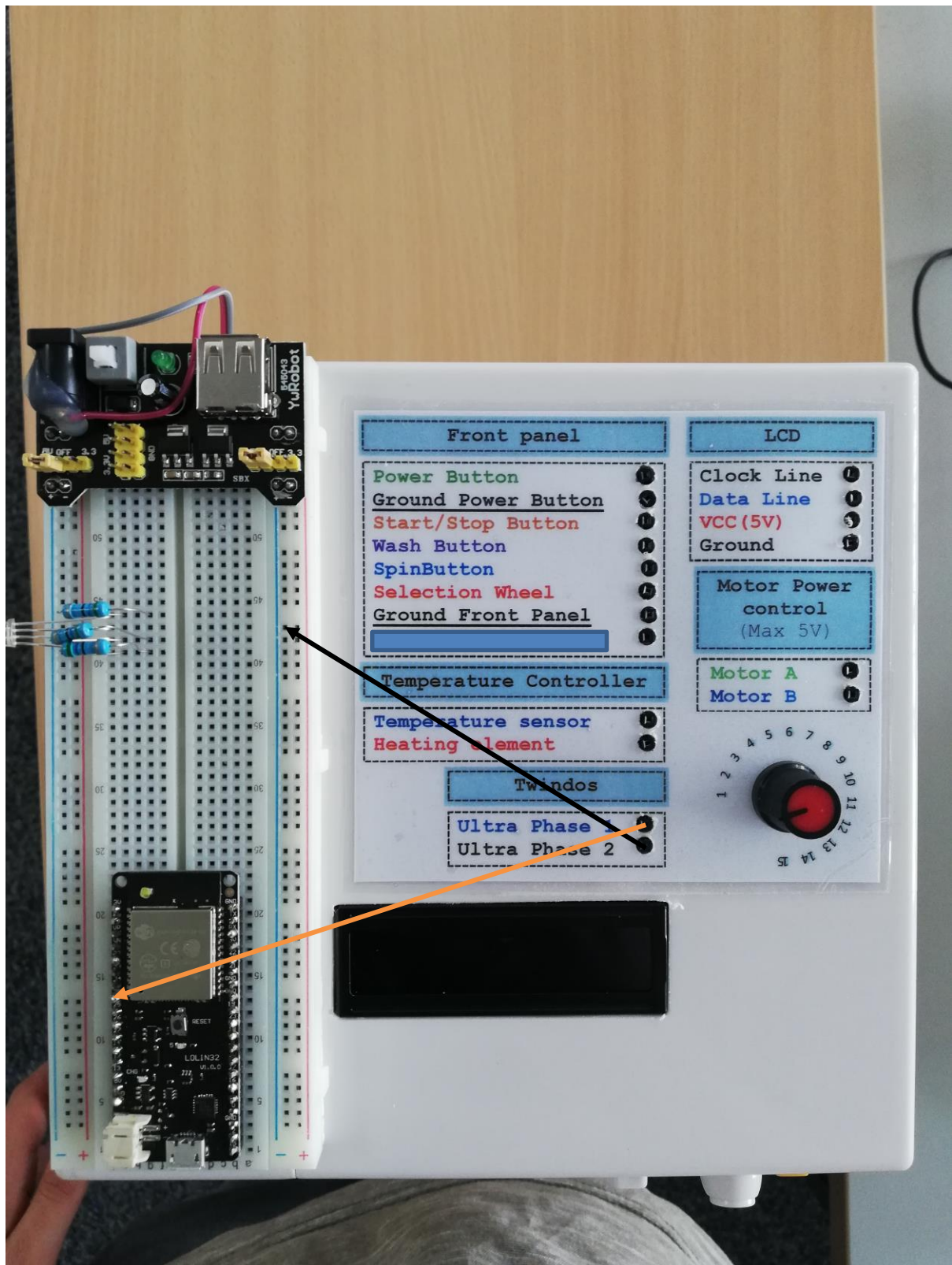


- 3) Firele pot fi conectate cum sunt in imagine sau diferit însă acestea trebuie specificate in inițializarea butonului. (LED-ul este conectat la pinii 25, 26, 27, iar firele negre reprezintă conectarea la ground)
- 4) Hint: Pentru o mai ușoară funcționare clasele *ControlButton* și *ControlPanel* au fost create. In interiorul claselor se vor găsi metodele pentru a detecta dacă un buton a fost apăsat și metodele de inițializare a acestora (în cazul în care se folosesc mai mult de două butoane din interfața mașinii atunci de preferat este să se folosească clasa *ControlPanel*)
- 5) În caz că în clasele mai sus menționate se vor afla //TBD acestea trebuie completate de participanți .
- 6) După ce codul a fost implementat acesta trebuie compilat și urcat pe microcontroler.
- 7) Pe final, trebuie să testăm dacă ceea ce a fost scris funcționează așa cum ne-am dorit și anume **dacă ușa este închisă și butonul de start a fost apăsat atunci ledul verde se va aprinde dacă nu ledul roșu se va aprinde. De testat ce se întâmplă dacă apăsăm butonul de start de mai multe ori și dacă roșu verde se aprinde atunci când ușa este deschisă.**



## Experiment 2 – Detectează dacă cartușele TwinDos sunt inserate.

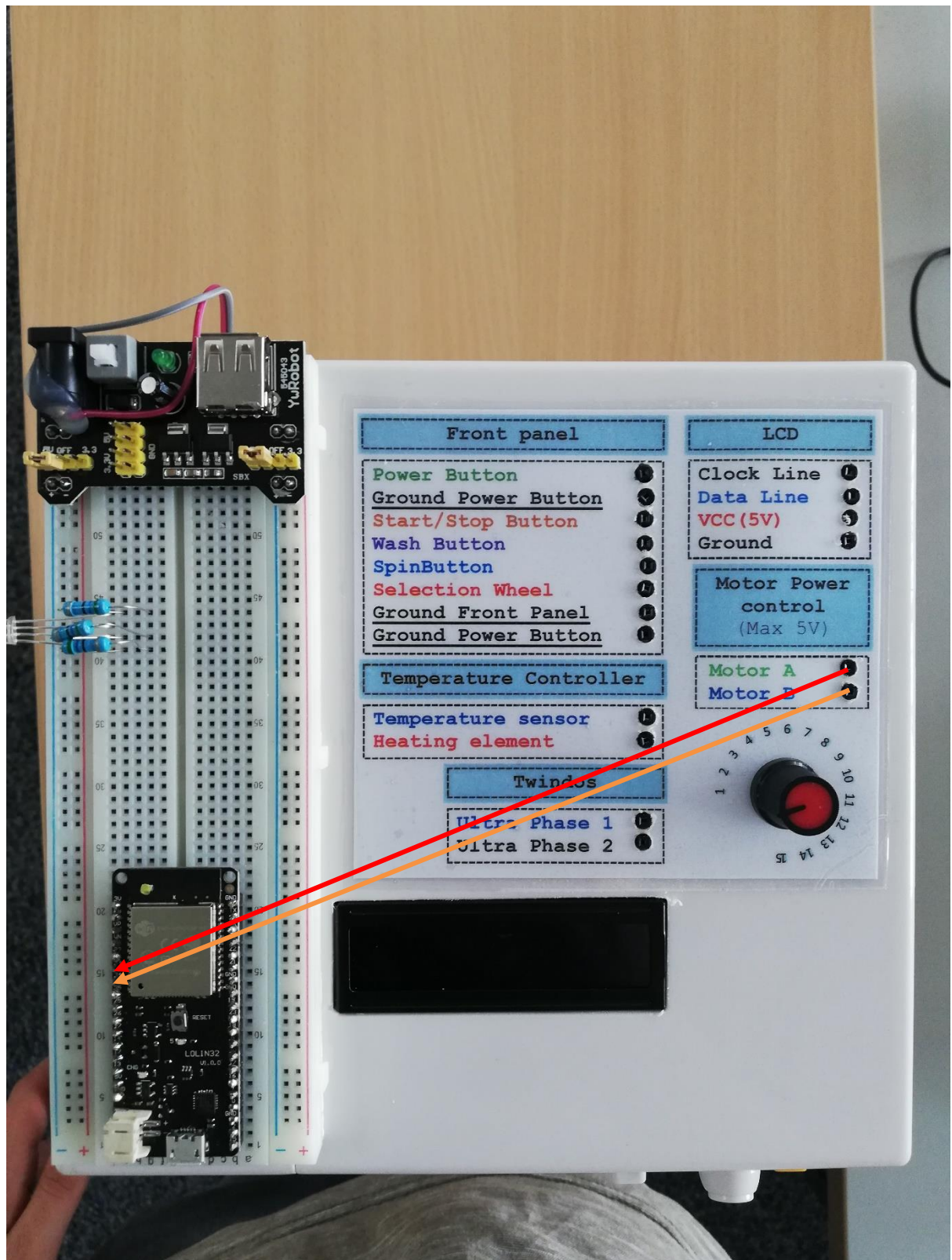
- 1) Ce ai nevoie: Cartușe TwinDos, fire, 2xReedSwitch sau Senzori de Prezență
- 2) Setup-ul arată cam așa(Cu experimentul anterior completat):



- 3) Firele pot fi conectate cum sunt in imagine sau diferit însă acestea trebuie specificate in inițializarea butonului/modulului (firul negru este conectat la ground, iar firul portocaliu la pinul 35).
- 4) Pentru utilizarea câtor mai puțini pini cei doi senzori de prezență au fost puși în serie, acest lucru fiind totodată și un dezavantaj pentru ca nu știm care senzor a fost introdus ci putem afla doar dacă amândoi senzorii sunt prezenți sau nu Cartușele TwinDos.
- 5) Pentru o organizare a codului mai bună, pentru Cartușele TwinDos s-a creat clasa TwinDos.
- 6) În caz ca în clasa de mai sus menționată se vor afla //TBD acestea trebuie completate de participanți
- 7) Ce este de făcut: De detectat dacă ambele cartușe TwinDos sunt inserate și dacă da, la apăsarea butonului de start ledul albastru să se aprindă. Din nou trebuie ca ușa să fie închisă și doar atunci ledul albastru se poate aprinde dacă nu ledul roșu se va aprinde.

## Experiment 3 – Să ne jucăm cu motorul

- 1) Ce ai nevoie: Motor, Sursă curent și punteH
- 2) Setup-ul arată cam așa(Cu experimentul anterior completat):

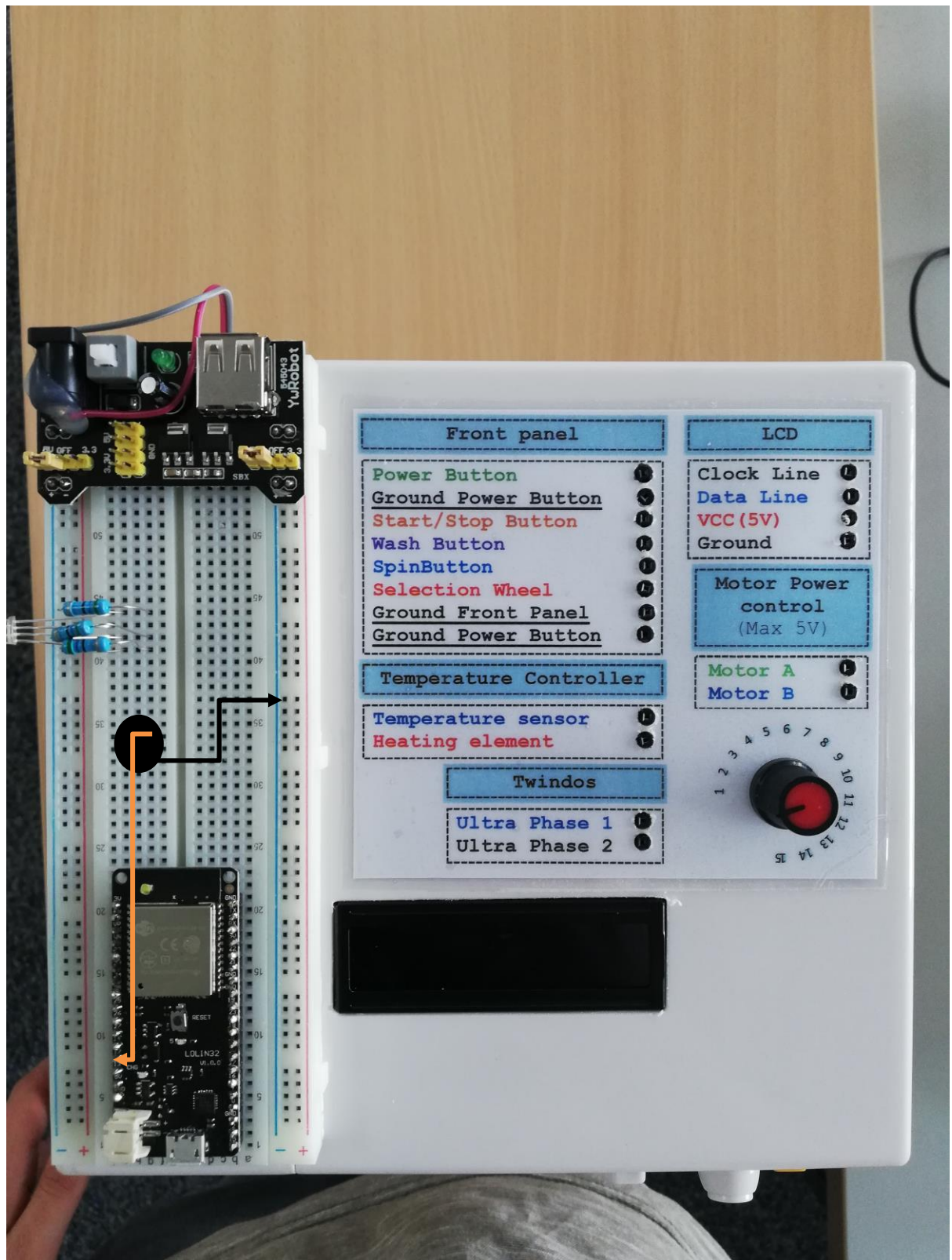




- 3) Firele pot fi conectate cum sunt in imagine sau diferit însă acestea trebuie specificate in inițializarea butonului/modulului (in imagine Motor A si B sunt conectate la pinii 33 si 34).
- 4) Pentru acest experiment ne vom folosi de capabilitățile PWM ale microcontrolerului.
- 5) Ce este de făcut: De rotit motorul într-un sens timp de 5 secunde și în celălalt sens timp de 3 secunde iar apoi să se oprească.
- 6) De folosit clasa MotorDriver
- 8) În caz ca în clasa de mai sus menționată se vor afla //TBD acestea trebuie completate de participanți
- 9) De testat codul

## Experiment 4 – Play a sound!

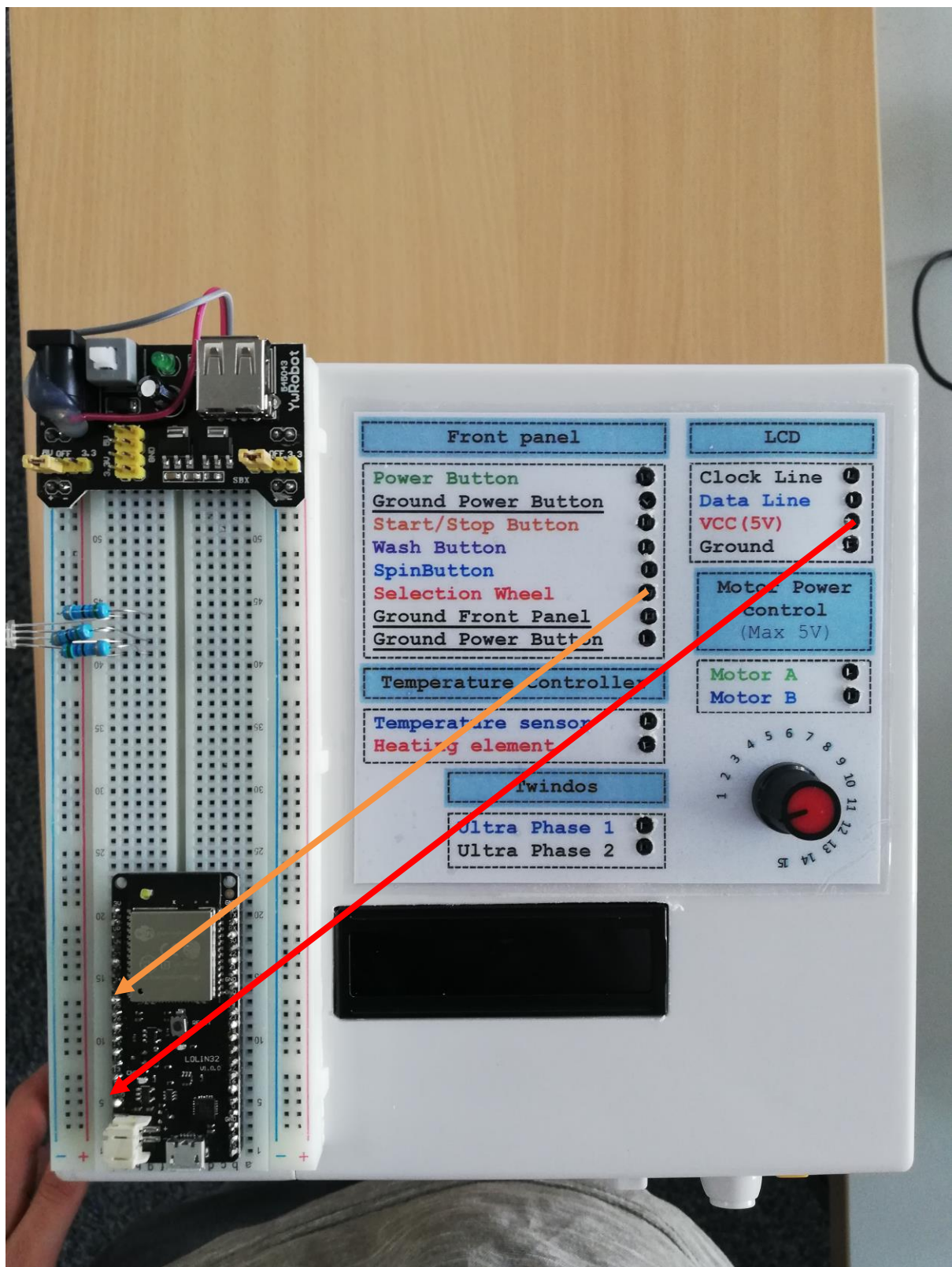
- 1) Ce ai nevoie: Buzzer si ESP
- 2) Setup-ul arată cam așa(Cu experimentul anterior completat):



- 3) De folosit modulul Sound.(firul negru este ground, iar cel portocaliu este predefinit pe pinul 12 )
- 4) În caz ca în modulul mai sus menționat se vor afla //TBD acestea trebuie completate de participanți
- 5) Ce este de făcut: atunci când ușa este închisă butonul de start a fost apăsat, cartușele TwinDos sunt prezente motorul se va roti 4 secunde dreapta, 4 va aștepta și 4 se va roti dreapta. La final după ce programul a terminat un sunet de notificare va fi redat.
- 6) Din nou se va face testarea de către participanți.

## Experiment 5 – Selectarea unui program cu ajutorul potențiometrului

- 1) Ce ai nevoie: Potențiometru si ESP
- 2) Setup-ul arată cam așa(Cu experimentul anterior completat, linia rosia va merge la 5v pe placa, iar linia portocalie la pinul 32):



- 3) Ce e de făcut: se vor defini 5 programe diferite de spălare. Cu ajutorul potențiometrului de 100k se vor selecta fiecare program în parte.
- 4) De testat.





## Experiment 6 – LCD 16x2

- 1) Ce ai nevoie: LCD 16x2
- 2) Setup-ul arată cam așa(Cu experimentul anterior completat, firul portocaliu conectat la pinul 22, iar firul rosu la pinul 21):

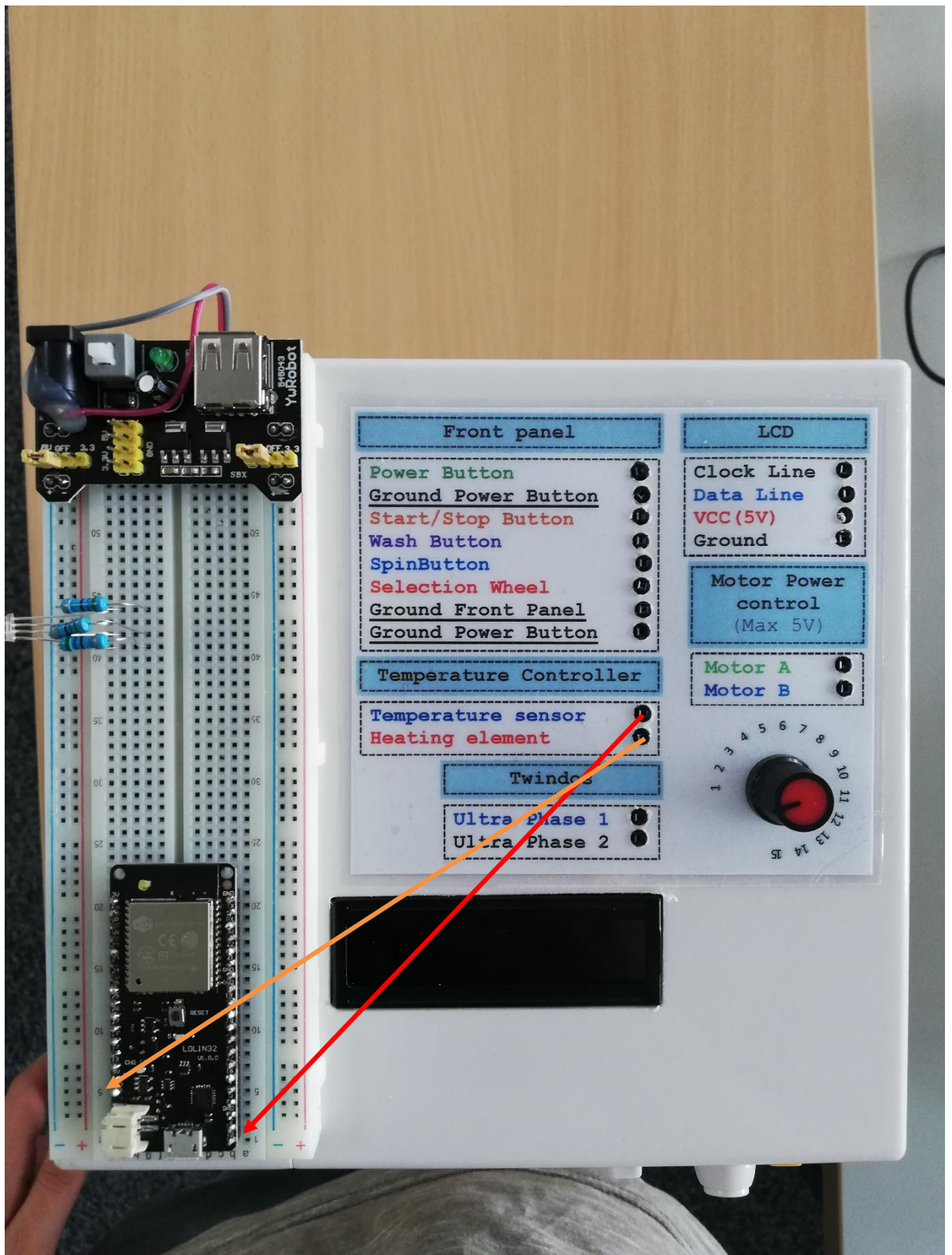


- 3) Conteaza LCD la ESP și apoi afișează programul curent care rulează.

- 4) Testează ca totul funcționează corect.

## Experiment 7 – Heater Module

- 1) Ce ai nevoie: Modul heater, ESP32
- 2) Setup-ul arată cam așa(Cu experimentul anterior completat, firul roșu la pinul 15 iar firul portocaliu la pinul 13 ):



- 3) Conecteaza heater la ESP și citește temperatura.
- 4) Testează ca totul funcționează corect.



