

Hướng Dẫn Sử Dụng Mạch Điều Khiển Động Cơ MakerEdu I2C Dual DC Motor Driver

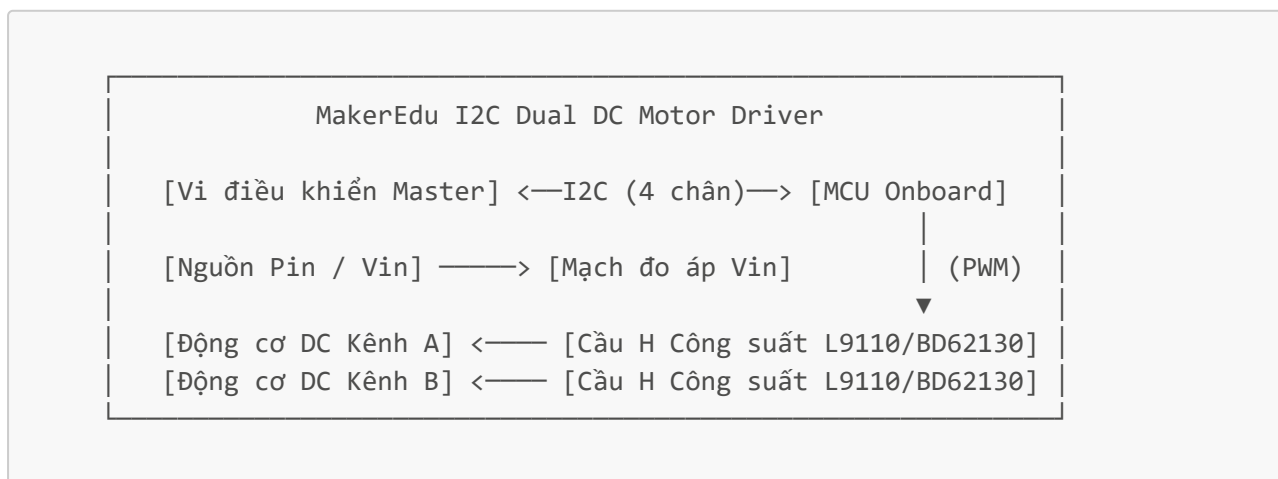
Phiên bản tài liệu: 1.1.0

Mã sản phẩm áp dụng:

- **MKE-M17:** Mạch MakerEdu I2C Dual DC Motor Driver L9110 (Silkscreen: **I2C DRIVER L9110**)
 - **MKE-M18:** Mạch MakerEdu I2C Dual DC Motor Driver BD62130 (Silkscreen: **I2C DRIVER BD62130**)
- Tương thích:** Arduino (Uno, Mega, Nano), ESP32, ESP8266, **BBC micro:bit (MakeCode Extension)**, Raspberry Pi, STM32

1. Giới thiệu Sản phẩm

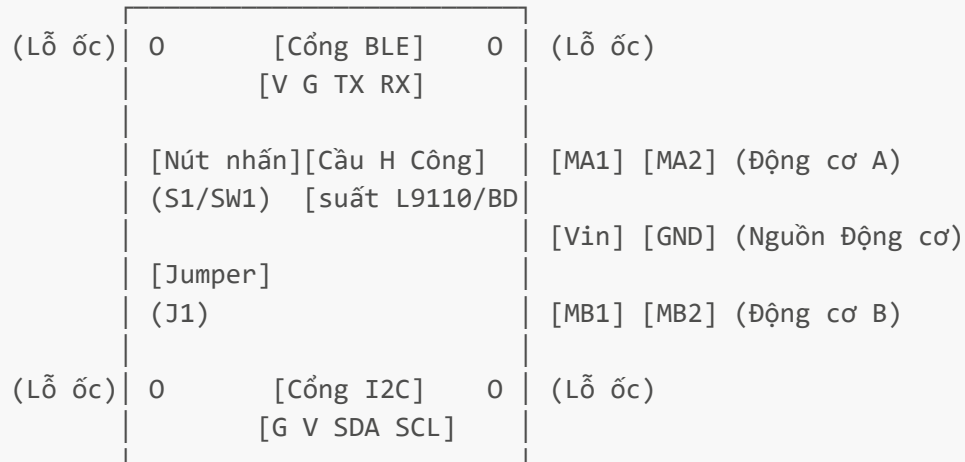
MakerEdu I2C Dual DC Motor Driver là dòng module điều khiển 2 động cơ DC thông minh trong hệ sinh thái MakerEdu, được trang bị vi điều khiển 32-bit ARM Cortex-M0+ tốc độ cao. Module đảm nhận toàn bộ việc phát xung PWM phần cứng, đảo chiều quay, hẹn giờ tự ngắt an toàn và đo điện áp nguồn, giao tiếp với vi điều khiển trung tâm (Master) chỉ qua **2 dây chuẩn I2C**.



So sánh Thông số Kỹ thuật: MKE-M17 (L9110) vs. MKE-M18 (BD62130)

Tiêu chí	MKE-M17 (L9110)	MKE-M18 (BD62130)
Mã sản phẩm (SKU)	MKE-M17	MKE-M18
Chữ in trên mạch (Silk)	I2C DRIVER L9110	I2C DRIVER BD62130
Điện áp Logic (VCC)	3.3V - 5.0V DC	3.3V - 5.0V DC
Điện áp Động cơ (Vin)	6.0V - 9.0V DC (Khuyến dùng 2S Li-ion / 6x Pin AA)	8.0V - 13.0V DC (Khuyến dùng 3S LiPo /Ắc quy 12V)
Dòng tải liên tục / Kênh	0.8 A	1.0 A
Dòng đỉnh (Peak) / Kênh	1.5 A	2.5 A (Tích hợp bảo vệ quá dòng OCP & quá nhiệt TSD)
Tần số PWM hỗ trợ	500 Hz – 2000 Hz	500 Hz – 20000 Hz
Ứng dụng tiêu biểu	Xe robot mini 2WD/4WD, động cơ vàng TT, động cơ N20	Robot kéo tải nặng, động cơ DC 370/520, khung gầm Mecanum

2. Giao diện & Sơ đồ Chân Ngoại vi



Chức năng Các Cổng & Ngoại vi Trên Mạch

1. Cổng I2C MakerEdu (Chuẩn 4-Pin):

- **GND**: Chân Mass chung.
- **VCC**: Nguồn nuôi mạch Logic (3.3V hoặc 5V từ Master).
- **SDA**: Đường truyền dữ liệu I2C (đã **tích hợp sẵn trở kéo 10kΩ**).
- **SCL**: Đường xung clock I2C (đã **tích hợp sẵn trở kéo 10kΩ**).

2. Cổng BLE / UART (Chuẩn 4-Pin):

- **VCC, GND, TX, RX**: Thiết kế chuẩn để kết nối module **MKE-M15 Bluetooth UART Module**.
- **Tính năng An toàn Vượt trội**: Khi kết nối điều khiển qua App **Dabble** trên smartphone, mạch liên tục giám sát tín hiệu truyền thông. Nếu xảy ra hiện tượng **mất sóng, ngoài vùng phủ sóng hoặc ngắt kết nối BLE**, MCU trên mạch sẽ **tự động kích hoạt cơ chế ngắt xung PWM, dừng khẩn cấp toàn bộ động cơ** để chống xe chạy mất kiểm soát.

3. Terminal Động cơ (Terminal Bắt vít Xanh):

- **MA1, MA2**: Cổng ra Motor A (kèm 2 LED báo chiều quay **MA1, MA2**).
- **MB1, MB2**: Cổng ra Motor B (kèm 2 LED báo chiều quay **MB1, MB2**).

4. Terminal Nguồn Cấp Động cơ:

- **Vin**: Cực dương nguồn nuôi động cơ (+6V đến +9V cho MKE-M17, +8V đến +13V cho MKE-M18). Có diode SS34 chống cắm ngược cực.
- **GND**: Cực âm nguồn động cơ.

5. Jumper / Công tắc Chọn Nguồn (J1):

- **J1 = ON**: Kích hoạt IC hạ áp LDO 5V trên mạch lấy nguồn từ **Vin** để hạ xuống 5V nuôi MCU và khối logic.

( Lưu ý: Chỉ nên để ON khi mạch hoạt động độc lập hoặc không kết nối chân VCC từ mạch I2C Master để tránh xung đột nguồn).

- **J1 = OFF (Mặc định):** Ngắt IC hạ áp LDO 5V. Lúc này toàn bộ phần điều khiển sẽ lấy nguồn từ cổng I2C (chân VCC của Master) hoặc cổng UART.

( Lưu ý: Khi cổng UART đang gắn module MKE-M15 Bluetooth, module này cũng tiêu thụ nguồn, do đó phần điều khiển bắt buộc phải lấy nguồn từ cổng I2C của Master).

6. Nút nhấn Onboard Đa năng (S1 / SW1):

- **Nhấn nhả 1 lần:** Chạy chu trình tự kiểm tra động cơ (Motor A quay tới/lùi, Motor B quay tới/lùi).
- **Nhấn giữ 4 giây:** Khôi phục cài đặt gốc và đưa địa chỉ I2C về mặc định **0x40** (Lưu ý: Mạch hiện tại chưa có đèn LED báo hiệu cho thao tác này, sau 4 giây địa chỉ sẽ tự động được khôi phục về **0x40**).

3. Sơ đồ Đấu Nối Phần Cứng

Kết nối với Arduino Uno (Chuẩn J1 = OFF)

Chân Arduino Uno	Chân Mạch MakerEdu Motor Driver	Chức năng
5V	VCC	Cấp nguồn Logic cho mạch
GND	GND	Mass chung
A4	SDA	Tín hiệu Data I2C
A5	SCL	Xung Clock I2C
Cực (+) Pin/Ắc quy	Vin (Terminal BẮT VÍT)	Nguồn động cơ (+6V..9V cho M17, +8V..13V cho M18)
Cực (-) Pin/Ắc quy	GND (Terminal BẮT VÍT)	Mass nguồn động cơ
Động cơ DC 1	MA1 / MA2	Kênh Motor A
Động cơ DC 2	MB1 / MB2	Kênh Motor B

4. Hướng Dẫn Cài Đặt Thư Viện Arduino

Bạn có thể cài đặt và sử dụng thư viện theo một trong hai cách:

Cách 1: Cài đặt thông qua Thư viện Hệ sinh thái **MKE_ONE** (Khuyến dùng)

1. Mở Arduino IDE, vào **Tools > Manage Libraries...** (hoặc nhấn **Ctrl + Shift + I**).
2. Tìm kiếm từ khóa **MKE_ONE** và nhấn **Install**.
3. Khi Arduino IDE hỏi có muốn cài đặt các thư viện phụ thuộc không, chọn **"Install All"** (IDE sẽ tự động tải luôn thư viện **MakerEdu_I2C_MotorDriver**).
4. Để mở các ví dụ mẫu chuẩn, vào menu:
 - **File > Examples > MKE_ONE > Module > MKE_M17_I2C_Motor_L9110** (cho mạch MKE-M17)
 - **File > Examples > MKE_ONE > Module > MKE_M18_I2C_Motor_BD62130** (cho mạch MKE-M18)

Cách 2: Cài đặt Thư viện Driver Độc lập

1. Mở Arduino IDE, vào **Tools > Manage Libraries....**
 2. Tìm kiếm từ khóa **MakerEdu I2C Motor Driver** và nhấn **Install**.
-

5. Cấu trúc Chương trình & Bảng Hàm API

Chương trình Cơ bản

```
#include <Wire.h>
#include <MKE_I2C_MotorDriver.h>

// Bạn có thể khai báo bằng MKE_I2C_MotorDriver, MKE_M17_MotorDriver hoặc
// MKE_M18_MotorDriver
MKE_M17_MotorDriver motorDriver;

void setup() {
  Serial.begin(9600);
  Wire.begin();           // Người dùng chủ động khởi tạo bus I2C
  motorDriver.begin();    // Liên kết module ở địa chỉ mặc định 0x40
}

void loop() {
  motorDriver.motorA_CW(200); // Motor A quay thuận tốc độ 200/255
  delay(1000);
  motorDriver.stopAll();     // Dừng tất cả động cơ
  delay(1000);
}
```

Bảng Tra cứu Toàn bộ Hàm API C++

Tên hàm	Tham số	Giá trị trả về	Chức năng
<code>begin(address, wire)</code>	<code>uint8_t address = 0x40, TwoWire &wire = Wire</code>	<code>void</code>	Cấu hình địa chỉ I2C và đối tượng Wire (Lưu ý: Gọi <code>Wire.begin()</code> trước trong <code>setup()</code>)
<code>motorA_CW(speed, duration_ms)</code>	<code>uint8_t speed (0-255), uint16_t duration_ms = 0</code>	<code>void</code>	Motor A quay thuận. <code>duration_ms = 0</code> là chạy liên tục, <code>> 0</code> là hẹn giờ tự dừng
<code>motorA_CCW(speed, duration_ms)</code>	<code>uint8_t speed (0-255), uint16_t duration_ms = 0</code>	<code>void</code>	Motor A quay ngược
<code>stopMotorA()</code>	Không	<code>void</code>	Dừng Motor A ngay lập tức
<code>motorB_CW(speed, duration_ms)</code>	<code>uint8_t speed (0-255), uint16_t duration_ms = 0</code>	<code>void</code>	Motor B quay thuận
<code>motorB_CCW(speed, duration_ms)</code>	<code>uint8_t speed (0-255), uint16_t duration_ms = 0</code>	<code>void</code>	Motor B quay ngược
<code>stopMotorB()</code>	Không	<code>void</code>	Dừng Motor B ngay lập tức
<code>stopAll()</code>	Không	<code>void</code>	Dừng đồng thời cả 2 kênh Motor
<code>getVin()</code>	Không	<code>uint32_t</code>	Đọc điện áp nguồn Vin (đơn vị: mV , ví dụ <code>8400</code> = 8.4V)
<code>setAddress(newAddress)</code>	<code>uint8_t newAddress (1-126)</code>	<code>void</code>	Đổi địa chỉ I2C mới, lưu vào EEPROM trên mạch
<code>getAddress()</code>	Không	<code>uint8_t</code>	Đọc địa chỉ I2C hiện tại trong EEPROM
<code>getModuleId()</code>	Không	<code>uint8_t</code>	Đọc mã định danh module (trả về 3)
<code>getFirmwareVersion()</code>	Không	<code>uint32_t</code>	Đọc phiên bản build firmware (ví dụ <code>260331</code>)
<code>setPwmFrequency(freq_hz)</code>	<code>uint32_t freq_hz</code>	<code>void</code>	Cài đặt tần số PWM (Khuyến dùng 500-2000 Hz)
<code>setPwmMA1(val), setPwmMA2(val), setPwmMB1(val), setPwmMB2(val)</code>	<code>uint8_t val (0-255)</code>	<code>void</code>	Xuất trực tiếp xung PWM thô ra từng chân cầu H

6. Các Ví dụ Ứng dụng Thực Tế

Ví dụ 1: Tăng giảm Tốc độ và Đảo chiều Cơ bản

```
#include <Wire.h>
#include <MKE_I2C_MotorDriver.h>

MKE_I2C_MotorDriver motorDriver;

void setup() {
  Serial.begin(9600);
  Wire.begin();
  motorDriver.begin();
}

void loop() {
  // Tăng tốc dần Motor A
  for (int spd = 0; spd <= 255; spd += 25) {
    motorDriver.motorA_CW(spd);
    delay(100);
  }
  delay(1000);
  motorDriver.stopMotorA();
  delay(500);

  // Đảo chiều Motor A
  motorDriver.motorA_CCW(200);
  delay(1500);
  motorDriver.stopAll();
  delay(2000);
}
```

Ví dụ 2: Hẹn Giờ Tự Dừng & An Toàn Watchdog trong Vòng Lặp PID

```
#include <Wire.h>
#include <MKE_I2C_MotorDriver.h>

MKE_I2C_MotorDriver motorDriver;

void setup() {
  Serial.begin(9600);
  Wire.begin();
  motorDriver.begin();
}

void loop() {
  // 1. Chạy tự hành: Quay thuận trong 2.5 giây rồi tự dừng
  motorDriver.motorA_CW(180, 2500); // Tốc độ 180, tự dừng sau 2500ms
  motorDriver.motorB_CW(180, 2500);

  // Vì điều khiển Master rảnh tay đọc cảm biến mà không bị block
  delay(4000);

  // 2. Vòng lặp PID có Watchdog: Gửi lệnh mỗi 100ms với timeout 200ms.
  // Nếu Arduino bị treo hoặc đứt dây I2C, sau 200ms động cơ tự dừng an toàn!
  for (int i = 0; i < 50; i++) {
    int speedA = 200;
    int speedB = 190;
    motorDriver.motorA_CW(speedA, 200);
    motorDriver.motorB_CW(speedB, 200);
    delay(100);
  }
  motorDriver.stopAll();
  delay(3000);
}
```

Ví dụ 3: Giám sát Nguồn Điện & Cảnh báo Pin Yếu

```
#include <Wire.h>
#include <MKE_I2C_MotorDriver.h>

MKE_I2C_MotorDriver motorDriver;

void setup() {
  Serial.begin(9600);
  Wire.begin();
  motorDriver.begin();
}

void loop() {
  uint32_t vin_mV = motorDriver.getVin();
  float vin_V = vin_mV / 1000.0;

  Serial.print(F("Dien ap Pin: "));
  Serial.print(vin_V, 2);
  Serial.println(F(" V"));

  // Bảo vệ pin 2S Li-ion (ngắt tải khi dưới 6.6V để chống hỏng cell)
  if (vin_mV > 0 && vin_mV < 6600) {
    Serial.println(F(">>> CANH BAO: Pin yeu! Dung dong co de bao ve Pin.
<<<"));
    motorDriver.stopAll();
    while (true) { delay(1000); } // Dừng chương trình
  }

  delay(1000);
}
```

Ví dụ 4: Ghép Nối Nhiều Mạch Motor Trên Cùng Bus I2C

```
#include <Wire.h>
#include <MKE_I2C_MotorDriver.h>

MKE_M17_MotorDriver banhTruoc; // Mạch 1 ở địa chỉ 0x40
MKE_M17_MotorDriver banhSau;   // Mạch 2 ở địa chỉ 0x41

void setup() {
    Wire.begin();
    banhTruoc.begin(0x40);
    banhSau.begin(0x41);

    // Điều khiển xe 4 bánh tiến tới
    banhTruoc.motorA_CW(200);
    banhTruoc.motorB_CW(200);
    banhSau.motorA_CW(200);
    banhSau.motorB_CW(200);
}

void loop() {}
```

Ví dụ 5: Điều Khiển Xe Robot Mecanum Đa Hướng (4WD / 2 Drivers)

```
#include <Wire.h>
#include <MKE_I2C_MotorDriver.h>

MKE_I2C_MotorDriver driverTruoc; // Địa chỉ 0x40 (FL: Motor A, FR: Motor B)
MKE_I2C_MotorDriver driverSau;   // Địa chỉ 0x41 (RL: Motor A, RR: Motor B)

// Hàm đặt tốc độ 4 bánh (-255 đến 255)
void set4Banh(int fl, int fr, int rl, int rr) {
    (fl >= 0) ? driverTruoc.motorA_CW(fl) : driverTruoc.motorA_CCW(-fl);
    (fr >= 0) ? driverTruoc.motorB_CW(fr) : driverTruoc.motorB_CCW(-fr);
    (rl >= 0) ? driverSau.motorA_CW(rl)   : driverSau.motorA_CCW(-rl);
    (rr >= 0) ? driverSau.motorB_CW(rr)   : driverSau.motorB_CCW(-rr);
}

void setup() {
    Wire.begin();
    driverTruoc.begin(0x40);
    driverSau.begin(0x41);
}

void loop() {
    // 1. Tiến thẳng (2s)
    set4Banh(200, 200, 200, 200); delay(2000);
    // 2. Dạt ngang sang trái (2s)
    set4Banh(-200, 200, 200, -200); delay(2000);
    // 3. Dạt ngang sang phải (2s)
    set4Banh(200, -200, -200, 200); delay(2000);
    // 4. Xoay tròn tại chỗ sang trái (1.5s)
    set4Banh(-180, 180, -180, 180); delay(1500);
    // 5. Dừng xe (2s)
    set4Banh(0, 0, 0, 0); delay(2000);
}
```

7. Hướng Dẫn Lập Trình Với BBC micro:bit (MakeCode Extension)

Mạch **MakerEdu I2C Dual DC Motor Driver** hỗ trợ hoàn hảo cho nền tảng **BBC micro:bit** thông qua Extension đồ họa trực quan trên Microsoft MakeCode.

1. Cài đặt Extension vào MakeCode

1. Mở trình duyệt truy cập: <https://makecode.microbit.org>
2. Tạo dự án mới (**New Project**).
3. Vào **Settings (Bánh răng) > Extensions**.
4. Dán đường link Extension:
https://github.com/Khuuxuanngoc/makeCode_MKE_I2C_Motor_Driver_extension_test và nhấn Enter để nạp.

2. Các Bài Học Hướng Dẫn Tương Tác Trực Tuyến (Interactive Tutorials)

Người học và giáo viên có thể nhấp trực tiếp vào các liên kết bên dưới để MakeCode tự động mở chế độ học tập từng bước (kèm gợi ý khối lệnh và tự nạp Extension):

-  **Bài 1: Điều khiển động cơ cơ bản & Hẹn giờ tự ngắt:**
 [Mở Tutorial 1 trên MakeCode](#)
-  **Bài 2: Đọc điện áp Pin Vin & Cảnh báo pin yếu:**
 [Mở Tutorial 2 trên MakeCode](#)
-  **Bài 3: Lập trình Xe Robot 2 Bánh Vi Sai (2WD Robot):**
 [Mở Tutorial 3 trên MakeCode](#)
-  **Bài 4: Lập trình Xe Robot Mecanum 8 Hướng (4WD / 2 Drivers):**
 [Mở Tutorial 4 trên MakeCode](#)

8. Giao thức Truyền nhận I2C Chuẩn (Raw Protocol)

Dành cho lập trình viên sử dụng trên các nền tảng khác như ESP-IDF (C), Raspberry Pi (Python [smbus2](#)), STM32 HAL:

1. Định dạng Gói Ghi (Master → Slave)

- **Độ dài cố định:** 6 bytes
- **Cấu trúc:** `[Region (0x05)] [ModeID (1 byte)] [Payload MSB] [Payload Byte 2] [Payload Byte 1] [Payload LSB]`
- **Thứ tự byte:** Big-Endian (Byte trọng số cao truyền trước).
- **Quy tắc Payload:**
 - Lệnh cơ bản: `payload = speed` (0–255).
 - Lệnh có hẹn giờ: `payload = ((uint32_t)duration_ms << 16) | (uint32_t)speed`.

2. Trình tự Đọc Dữ liệu (Master ⇌ Slave)

1. Gửi gói Write 6 byte với **ModeID** cần đọc (`payload = 0`).
2. Tạo độ trễ tối thiểu **200 µs** (`delayMicroseconds(200)`) để Slave MCU chuẩn bị dữ liệu.

3. Gửi lệnh I2C Read đọc **4 bytes** (`requestFrom(address, 4)`).
4. Ghép 4 bytes nhận được thành số nguyên 32-bit Big-Endian.

3. Bảng Mã Lệnh I2C

ModelID (Thập phân)	Tên lệnh	Loại	Định dạng Payload	Ý nghĩa
106	Set_MA_CW	Write	<code>(duration_ms << 16) \ speed</code>	Motor A quay thuận (0–255)
107	Set_MA_CCW	Write	<code>(duration_ms << 16) \ speed</code>	Motor A quay ngược
110	Set_MA_STOP	Write	<code>0</code>	Dừng ngay Motor A
108	Set_MB_CW	Write	<code>(duration_ms << 16) \ speed</code>	Motor B quay thuận
109	Set_MB_CCW	Write	<code>(duration_ms << 16) \ speed</code>	Motor B quay ngược
111	Set_MB_STOP	Write	<code>0</code>	Dừng ngay Motor B
112	Get_VIN	Read	Trả về: <code>uint32_t</code> (mV)	Đọc điện áp nguồn Vin
104	Set_MOTOR_FREQ	Write	<code>frequency_hz</code>	Đặt tần số PWM (500–2000Hz)
1	SetAddress	Write	<code>new_address</code> (1–126)	Đổi địa chỉ I2C của mạch
10	GetAddress	Read	Trả về: <code>uint8_t</code> address	Đọc địa chỉ lưu trong EEPROM
2	Get_ID_Module	Read	Trả về: <code>uint8_t</code> (3)	Đọc mã nhận diện Module
4	Get_FW_Version	Read	Trả về: <code>uint32_t</code>	Đọc phiên bản ngày build FW

9. Điều Khiển Gamepad Qua Ứng Dụng Dabble với Module MKE-M15

Mạch tích hợp sẵn bộ giải mã giao thức của ứng dụng **Dabble** (STEMpedia) điều khiển qua Bluetooth:

1. **Kết nối Phần cứng:** Cắm module **MKE-M15 Bluetooth UART Module** vào **Cổng BLE** (**VCC**, **GND**, **TX**, **RX**) trên mạch driver.
2. **Kết nối Ứng dụng:** Mở ứng dụng **Dabble** trên smartphone và kết nối Bluetooth với module MKE-M15.
3. **Mở Giao diện Gamepad** (hỗ trợ cả Digital Mode và Joystick Mode):
 - **Nút START (Kích hoạt điều khiển):** Sau khi mở Gamepad, **nhấn nút START** để mạch driver nhận biết bắt đầu kết nối ("Connect") và cho phép nhận lệnh điều khiển động cơ.
 - **Nút SELECT (Tạm ngắt điều khiển):** Nhấn nút **SELECT** để tạm khóa/ngắt lệnh điều khiển, giúp xe đứng yên an toàn khi tạm nghỉ mà không cần ngắt kết nối Bluetooth.
 - **Nút VUÔNG (Square):** Phanh/dừng ngay lập tức cả 2 kênh động cơ.
 - **⬆️ Nút LÊN / Gạt Joystick Lên:** Cả 2 động cơ chạy Tiến (Forward).
 - **⬇️ Nút XUỐNG / Gạt Joystick Xuống:** Cả 2 động cơ chạy Lùi (Backward).
 - **⬅️ Nút TRÁI (Left):** Xoay trái vi sai (Motor A lùi, Motor B tiến).
 - **➡️ Nút PHẢI (Right):** Xoay phải vi sai (Motor A tiến, Motor B lùi).
 - **🕹️ Điều khiển Joystick:** Tự động điều chỉnh góc lái và tốc độ mượt mà tỉ lệ theo tọa độ trục X/Y (-7.0 ~ +7.0).
4. **🛡️ Cơ chế Bảo vệ Mất sóng (Failsafe):** Nếu điện thoại bị mất kết nối Bluetooth, ngoài vùng phủ sóng hoặc thoát ứng dụng Dabble, mạch sẽ tự động ngắt xung PWM và dừng toàn bộ động cơ ngay lập tức, đảm bảo an toàn tuyệt đối.

10. Lưu Ý Quan Trọng & Xử Lý Sự Cố

[!IMPORTANT] Khởi tạo `Wire.begin()` trong Sketch: Thư viện **MKE_I2C_MotorDriver** **không tự ý gọi `Wire.begin()`** bên trong. Người dùng luôn khởi tạo **`Wire.begin()`** trong hàm **`setup()`**. Điều này giúp bạn linh hoạt đổi chân I2C (như trên ESP32: **`Wire.begin(SDA, SCL)`**) hoặc tăng tốc độ giao tiếp (**`Wire.setClock(400000)`**).

[!WARNING] Cấu hình Nguồn J1 & Điện áp Vin:

- Luôn để **J1 = OFF** khi sử dụng chung với vi điều khiển Master (Arduino/ESP32) và module MKE-M15.
- **MKE-M17 (L9110):** Điện áp Vin định mức **6V - 9V DC**.
- **MKE-M18 (BD62130):** Điện áp Vin định mức **8V - 13V DC**.
- Không làm ngắn mạch 2 cọc ra động cơ.

[!TIP] Khôi phục Cài đặt Gốc Nhanh: Nếu quên địa chỉ I2C đã đổi, chỉ cần **nhấn giữ nút nhấn onboard (S1/SW1) trong 4 giây**. Mạch sẽ tự động xóa bộ nhớ EEPROM và khôi phục về địa chỉ mặc định **0x40** (Lưu ý: Hiện tại phần cứng chưa có đèn LED báo hiệu cho thao tác này).