



I've got a board with RP2040 controller, it has a single button on it, it is connected in serial by UART to 2 motors. can you please provide a program in Arduino IDE that will allow me to control these motors with the one button. the idea is that a long press of the button (i.e. more than one second) will toggle between the boards and show an associated LED, while a short press of less than half a second will toggle between motor actions (left, right, stop), on the selected motor

Here is a sketch for Arduino IDE to control two motors connected via UART to an RP2040 board with one button and an LED indication. The program supports:

- A long press of the button (> 1 second) toggles which motor is selected and lights the associated LED.
- A short press (< 0.5 seconds) cycles through motor actions (left, right, stop) on the selected motor.
- The motors are controlled by sending commands through UART.

This code assumes you have RX and TX pins defined for UART, one button pin, and two LEDs indicating motor selection.

```
#include <Arduino.h>

// Pins
const int buttonPin = 2;           // Button input pin
const int ledMotor1Pin = 10;       // LED for motor 1 selected
const int ledMotor2Pin = 11;       // LED for motor 2 selected

// UART pins for motors (use appropriate pins & Serial instance for RP2040)
#define MOTOR_SERIAL Serial1

// Button press timing thresholds (ms)
const unsigned long longPressTime = 1000;
const unsigned long shortPressTime = 500;

enum MotorAction {STOP, LEFT, RIGHT};
```

```

// Variables
int selectedMotor = 1; // Motor 1 or 2
MotorAction motorActions[^1_2] = {STOP, STOP};
bool lastButtonState = HIGH;
unsigned long buttonPressStart = 0;
bool buttonPressed = false;

void setup() {
    pinMode(buttonPin, INPUT_PULLUP);
    pinMode(ledMotor1Pin, OUTPUT);
    pinMode(ledMotor2Pin, OUTPUT);

    // Start UART for motor control at 115200 baud
    MOTOR_SERIAL.begin(115200);

    updateLeds();
}

void updateLeds() {
    digitalWrite(ledMotor1Pin, selectedMotor == 1 ? HIGH : LOW);
    digitalWrite(ledMotor2Pin, selectedMotor == 2 ? HIGH : LOW);
}

void sendMotorCommand(int motor, MotorAction action) {
    // Example command format: M1L (Motor 1 Left), M2S (Motor 2 Stop)
    char cmd[^1_5];
    cmd[^1_0] = 'M';
    cmd[^1_1] = '0' + motor;
    switch (action) {
        case LEFT: cmd[^1_2] = 'L'; break;
        case RIGHT: cmd[^1_2] = 'R'; break;
        case STOP: cmd[^1_2] = 'S'; break;
    }
    cmd[^1_3] = '\n';
    cmd[^1_4] = 0;
    MOTOR_SERIAL.print(cmd);
}

void loop() {
    bool currentButtonState = digitalRead(buttonPin);

    if (lastButtonState == HIGH && currentButtonState == LOW) {
        // Button just pressed
        buttonPressStart = millis();
        buttonPressed = true;
    }

    if (lastButtonState == LOW && currentButtonState == HIGH && buttonPressed) {
        // Button just released
        unsigned long pressDuration = millis() - buttonPressStart;

        if (pressDuration > longPressTime) {
            // Long press - toggle motor selection
            selectedMotor = (selectedMotor == 1) ? 2 : 1;
            updateLeds();
        } else if (pressDuration < shortPressTime) {

```

```

        // Short press - cycle motor action
        motorActions[selectedMotor - 1] = MotorAction((motorActions[selectedMotor - 1] + 1) % 3);
        sendMotorCommand(selectedMotor, motorActions[selectedMotor - 1]);
    }

    buttonPressed = false;
}

lastButtonState = currentButtonState;
}

```

This sketch uses one button with debounced press detection and timing to distinguish short and long presses. Motor commands are formatted as ASCII strings sent over UART for control. LEDs show which motor is currently selected.

You should connect motor UART lines to your RP2040's serial pins and verify your actual motor commands match what your motor controllers expect. Adjust pins and UART instance as needed for your specific RP2040 board and wiring.

This example is compatible with Arduino IDE and basic RP2040 serial usage for motor control through UART.^[1] ^[2]

**

RP2040 Controller Code

This Arduino sketch for your RP2040 uses the specified pins (RGB LED on 26-28, status LED on 2, button on 3). Long press (>1s) toggles motor selection (red/green LEDs), short press (<0.5s) cycles actions (left/right/stop) on selected motor and shows action via blue LED blinks.

Commands sent via Serial1 (pins 0/1 by default).^[8] ^[9]

```

int red = 28, green = 27, blue = 26, led = 2, button = 3;
int selectedMotor = 1; // 1 or 2
int motorAction[2] = {0, 0}; // 0=stop, 1=left, 2=right
bool lastButton = HIGH;
unsigned long pressStart = 0;
bool pressed = false;

void setup() {
    pinMode(button, INPUT_PULLUP);
    pinMode(led, OUTPUT);
    pinMode(red, OUTPUT); pinMode(green, OUTPUT); pinMode(blue, OUTPUT);
    Serial1.begin(115200); // UART to motors
    updateLEDs();
}

void updateLEDs() {
    digitalWrite(red, selectedMotor == 1 ? HIGH : LOW);
    digitalWrite(green, selectedMotor == 2 ? HIGH : LOW);
    digitalWrite(led, HIGH);
}

```

```

void showAction(int action) {
  for(int i = 0; i < action + 1; i++) {
    digitalWrite(blue, HIGH); delay(100); digitalWrite(blue, LOW); delay(100);
  }
  delay(200);
}

void sendCommand(int motor, int action) {
  Serial1.print("M"); Serial1.print(motor);
  if(action == 1) Serial1.print("L");
  else if(action == 2) Serial1.print("R");
  else Serial1.print("S");
  Serial1.println();
}

void loop() {
  bool btn = digitalRead(button);
  if(lastButton == HIGH && btn == LOW) { pressStart = millis(); pressed = true; }
  if(lastButton == LOW && btn == HIGH && pressed) {
    unsigned long duration = millis() - pressStart;
    if(duration > 1000) {
      selectedMotor = (selectedMotor == 1) ? 2 : 1;
      updateLEDs();
    } else if(duration < 500) {
      motorAction[selectedMotor-1] = (motorAction[selectedMotor-1] + 1) % 3;
      sendCommand(selectedMotor, motorAction[selectedMotor-1]);
      showAction(motorAction[selectedMotor-1]);
    }
    pressed = false;
  }
  lastButton = btn;
}

```

ATtiny1614 Motor Board Code

Use megaTinyCore in Arduino IDE for ATtiny1614. Each motor board receives UART commands (M1L, M2R, M1S etc.) on its RX pin and controls its motor. Connect TX/RX/GND between RP2040 Serial1 and motor board UART. Assumes motor driver pins on ATtiny1614 (adjust as needed).^{[10] [11]}

```

// ATtiny1614 Motor Board (one per motor, change motorNum)
#define motorNum 1 // Change to 2 for second motor
int pwmA = 3, pwmB = 4, dirA = 5, dirB = 6; // Motor driver pins

void setup() {
  pinMode(pwmA, OUTPUT); pinMode(pwmB, OUTPUT);
  pinMode(dirA, OUTPUT); pinMode(dirB, OUTPUT);
  Serial.begin(115200); // UART from RP2040
  stopMotor();
}

void loop() {
  if(Serial.available() >= 3) {
    if(Serial.read() == 'M' && Serial.read() == ('0' + motorNum)) {

```

```

    char cmd = Serial.read();
    if(cmd == 'L') left();
    else if(cmd == 'R') right();
    else if(cmd == 'S') stopMotor();
  }
  while(Serial.available()) Serial.read(); // Clear buffer
}
}

void left() {
  digitalWrite(dirA, HIGH); digitalWrite(dirB, LOW);
  analogWrite(pwmA, 200); analogWrite(pwmB, 0);
}

void right() {
  digitalWrite(dirA, LOW); digitalWrite(dirB, HIGH);
  analogWrite(pwmA, 0); analogWrite(pwmB, 200);
}

void stopMotor() {
  analogWrite(pwmA, 0); analogWrite(pwmB, 0);
}

```

Upload RP2040 code first, then duplicate ATtiny code for each motor board (change `motorNum`). Wire RP2040 TX (pin 0) to both motor RX, or use separate UARTs if needed. Test UART at 115200 baud with common ground. [\[12\]](#) [\[13\]](#)

✱✱

Updated ATtiny1614 Motor Board Code (DRV8825)

DRV8825 stepper drivers require STEP/DIR control. Connect ATtiny1614 pins to DRV8825: STEP (pin 3), DIR (pin 4), EN (pin 5 to GND or LOW to enable), SLEEP/RESET (to VCC), GND logic to ATtiny GND. Set microstepping jumpers and Vref current limit per motor specs. Motor coils to A1/A2/B1/B2. [\[18\]](#) [\[19\]](#) [\[20\]](#)

```

// ATtiny1614 + DRV8825 Motor Board (duplicate for motor 2, change motorNum)
#define motorNum 1 // Set to 2 for second motor
int stepPin = 3;   // DRV8825 STEP
int dirPin = 4;    // DRV8825 DIR
int enablePin = 5; // DRV8825 EN (LOW=enabled)

unsigned long lastStepTime = 0;
const unsigned long stepDelay = 1000; // Adjust speed (us between steps)

void setup() {
  pinMode(stepPin, OUTPUT);
  pinMode(dirPin, OUTPUT);
  pinMode(enablePin, OUTPUT);
  digitalWrite(enablePin, LOW); // Enable driver
  Serial.begin(115200);         // UART from RP2040
  stopMotor();
}

```

```

void loop() {
  if(Serial.available() >= 3) {
    if(Serial.read() == 'M' && Serial.read() == ('0' + motorNum)) {
      char cmd = Serial.read();
      if(cmd == 'L') left();
      else if(cmd == 'R') right();
      else if(cmd == 'S') stopMotor();
    }
    while(Serial.available()) Serial.read(); // Clear buffer
  }
}

void left() {
  digitalWrite(dirPin, LOW); // Direction LEFT
  runMotor();
}

void right() {
  digitalWrite(dirPin, HIGH); // Direction RIGHT
  runMotor();
}

void runMotor() {
  digitalWrite(enablePin, LOW);
  while(true) { // Continuous stepping until new command
    unsigned long now = micros();
    if(now - lastStepTime >= stepDelay) {
      digitalWrite(stepPin, HIGH);
      delayMicroseconds(5);
      digitalWrite(stepPin, LOW);
      lastStepTime = now;
    }
    if(Serial.available()) break; // Check for new command
  }
}

void stopMotor() {
  digitalWrite(enablePin, HIGH); // Disable driver (holds position)
}

```

RP2040 Code (Unchanged)

The previous RP2040 code works as-is with these motor boards. Serial1 TX (GPIO0) connects to both motor RX pins (with common GND). Commands "M1L", "M2R", "M1S" etc. control direction/speed via STEP pulses. [\[21\]](#) [\[22\]](#)

Wiring Summary:

DRV8825	ATtiny1614	
STEP	Pin 3	
DIR	Pin 4	

DRV8825	ATtiny1614	
EN	Pin 5	
GND LOG	GND	
RX	UART RX	[18] [19]

Upload megaTinyCore board package for ATtiny1614. Adjust `stepDelay` for speed, set DRV8825 Vref (1.77V for 1.5A), and verify SLEEP/RESET tied high. [\[20\]](#)



Updated ATtiny1614 Motor Board Code (Motors A/B)

Based on your working DRV8825 code, this version receives "A" or "B" commands from RP2040 (e.g., "A" for Motor A Left/Right/Stop). Uses your exact pins (STEP=0, DIR=1), baud rate (57600), and 200-step movement with 1ms timing. Duplicate for Motor B, changing `motorID`. [\[29\]](#)
[\[30\]](#)

```
// ATtiny1614 Motor Board - Motor A (change motorID to 'B' for second motor)
const char motorID = 'A'; // 'A' for Motor A, 'B' for Motor B
const int dirPin = 1;     // DIR pin
const int stepPin = 0;    // STEP pin

void setup() {
  Serial.begin(57600);
  pinMode(dirPin, OUTPUT);
  pinMode(stepPin, OUTPUT);
}

void loop() {
  if(Serial.available() > 0) {
    char cmd = Serial.read();
    if(cmd == motorID) {
      while(Serial.available() == 0) delay(10); // Wait for direction command
      char dir = Serial.read();
      if(dir == 'L' || dir == 'l') left();
      else if(dir == 'R' || dir == 'r') right();
      else if(dir == 'S' || dir == 's') {} // Stop (do nothing)
    }
    while(Serial.available()) Serial.read(); // Clear buffer
  }
}

void right() {
  digitalWrite(dirPin, HIGH);
  for(int i = 0; i < 200; i++) {
    digitalWrite(stepPin, HIGH);
    delayMicroseconds(1000);
    digitalWrite(stepPin, LOW);
    delayMicroseconds(1000);
  }
}
```

```

void left() {
  digitalWrite(dirPin, LOW);
  for(int i = 0; i < 200; i++) {
    digitalWrite(stepPin, HIGH);
    delayMicroseconds(1000);
    digitalWrite(stepPin, LOW);
    delayMicroseconds(1000);
  }
}

```

Updated RP2040 Controller Code

Sends "A" or "B" followed by direction (L/R/S). Red LED = Motor A, Green LED = Motor B, Blue blinks show action (1=Left, 2=Right, 3=Stop).[\[31\]](#) [\[32\]](#)

```

int red = 28, green = 27, blue = 26, led = 2, button = 3;
char selectedMotor = 'A'; // 'A' or 'B'
int motorAction[^4_2] = {0, 0}; // 0=stop, 1=left, 2=right (index 0=A, 1=B)
bool lastButton = HIGH;
unsigned long pressStart = 0;
bool pressed = false;

void setup() {
  pinMode(button, INPUT_PULLUP);
  pinMode(led, OUTPUT); pinMode(red, OUTPUT); pinMode(green, OUTPUT); pinMode(blue, OUTPUT);
  Serial1.begin(57600); // Match ATtiny baud rate
  updateLEDs();
}

void updateLEDs() {
  digitalWrite(red, selectedMotor == 'A' ? HIGH : LOW);
  digitalWrite(green, selectedMotor == 'B' ? HIGH : LOW);
  digitalWrite(led, HIGH);
}

void showAction(int action) {
  for(int i = 0; i < action + 1; i++) {
    digitalWrite(blue, HIGH); delay(100); digitalWrite(blue, LOW); delay(100);
  }
  delay(200);
}

void sendCommand(char motor, int action) {
  Serial1.print(motor); // A or B
  if(action == 1) Serial1.println('L');
  else if(action == 2) Serial1.println('R');
  else Serial1.println('S');
}

void loop() {
  bool btn = digitalRead(button);
  if(lastButton == HIGH && btn == LOW) { pressStart = millis(); pressed = true; }
  if(lastButton == LOW && btn == HIGH && pressed) {

```



```

    unsigned long duration = millis() - pressStart;
    if(duration > 1000) {
        selectedMotor = (selectedMotor == 'A') ? 'B' : 'A';
        updateLEDs();
    } else if(duration < 500) {
        int motorIdx = (selectedMotor == 'A') ? 0 : 1;
        motorAction[motorIdx] = (motorAction[motorIdx] + 1) % 3;
        sendCommand(selectedMotor, motorAction[motorIdx]);
        showAction(motorAction[motorIdx]);
    }
    pressed = false;
}
lastButton = btn;
}

```

Wiring: RP2040 Serial1 TX (GPIO0) → Both ATtiny RX pins + common GND. Each ATtiny gets its own DRV8825 on STEP/DIR pins 0/1.^[33]



1. <https://community.simplefoc.com/t/arduino-nano-rp2040-support/7098>
2. <https://forum.arduino.cc/t/operating-nema17-stepper-motor-at-higher-velocity-with-tmc2209-based-on-rp2040-pi-pico/1273070>
3. <https://fabacademy.org/2024/labs/oulu/students/joni-rajala/assignments/steps/w07-programming/>
4. https://www.reddit.com/r/arduino/comments/1m7y9xm/controlling_2_rp2040_with_a_arduino_mega2560_board/
5. <https://forum.arduino.cc/t/rp2040-connect-controlling-servo/1103483>
6. <https://joachimweise.github.io/post/2020-05-17-serial-connection/>
7. <https://learn.adafruit.com/rp2040-arduino-with-the-earlephilhower-core/overview>
8. https://www.youtube.com/watch?v=WDtSjs0_2z4
9. <https://forum.arduino.cc/t/using-ttl-for-serial-communication-in-rp2040-in-arduino-ide/1272814>
10. <https://forum.arduino.cc/t/attiny-series-1-u-s-art/1223124>
11. <https://www.electronics-lab.com/project/using-new-attiny-processors-arduino-ide-attiny412-attiny1614-attiny3216-attiny1616-attiny3217/>
12. <https://fabacademy.org/2024/labs/charlotte/students/angelina-yang/assignments/week14/>
13. <https://forum.arduino.cc/t/uart-communication-between-rp2040-and-uno/1103529>
14. <http://www.technoblogy.com/show?RPY>
15. <https://www.youtube.com/watch?v=MUT6ZubKS3w>
16. <https://github.com/Arduino-IRremote/Arduino-IRremote>
17. <https://forum.arduino.cc/t/arduino-nano-rp2040-connect-breaks-with-sample-sketch/1266052>
18. <https://mytectutor.com/drv8825-stepper-motor-driver-with-arduino/>
19. <https://lastminuteengineers.com/drv8825-stepper-motor-driver-arduino-tutorial/>
20. <https://www.makerguides.com/drv8825-stepper-motor-driver-arduino-tutorial/>
21. <https://forum.arduino.cc/t/dvr8825-and-attiny85/543792>
22. <https://forum.arduino.cc/t/uart-communication-between-rp2040-and-uno/1103529>

23. <https://www.youtube.com/watch?v=AfDqchJLPxU>
24. https://www.youtube.com/watch?v=J-8_txDnS3w
25. <https://fabacademy.org/2024/labs/kochi/students/kalyani-rk/assignments/WEEK 9.html>
26. <https://www.youtube.com/watch?v=NrKMpnL9iCY>
27. <https://fabacademy.org/2024/labs/kochi/students/laxmi-renjith/assignments/week09.html>
28. <https://www.instructables.com/Controll-a-Stepper-Motor-With-the-DRV8825/>
29. <https://mytectutor.com/drv8825-stepper-motor-driver-with-arduino/>
30. <https://lastminuteengineers.com/drv8825-stepper-motor-driver-arduino-tutorial/>
31. https://www.youtube.com/watch?v=WDtSjs0_2z4
32. <https://forum.arduino.cc/t/using-ttl-for-serial-communication-in-rp2040-in-arduino-ide/1272814>
33. <https://forum.arduino.cc/t/uart-communication-between-rp2040-and-uno/1103529>