Arduino Code

#define BT\_TX 3

#define BT\_RX 4

#include "SoftwareSerial.h"

SoftwareSerial BTSerial(BT\_TX, BT\_RX); // Maker UNO RX, TX

#define BUTTON 2

#define PIEZO 8

#define L293N\_ENA 5

#define L293N\_ENB 6

#define L293N\_IN1 7

#define L293N\_IN2 9

#define L293N\_IN3 10

#define L293N\_IN4 11

#define NOTE\_G4 392

#define NOTE\_C5 523

#define NOTE\_G5 784

#define NOTE\_C6 1047

int btConnect[] = {NOTE\_G5, NOTE\_C6};

int btConnectNoteDurations[] = {12, 8};

int btDisconnect[] = {NOTE\_C5, NOTE\_G4};

int btDisconnectNoteDurations[] = {12, 8};

#define playBtConnectMelody() playMelody(btConnect, btConnectNoteDurations, 2)

#define playBtDisconnectMelody() playMelody(btDisconnect, btDisconnectNoteDurations, 2)

boolean BTConnect = false;

char inChar;

String inString;

void setup()

{

pinMode(BUTTON, INPUT\_PULLUP);

pinMode(L293N\_ENA, OUTPUT);

pinMode(L293N\_ENB, OUTPUT);

pinMode(L293N\_IN1, OUTPUT);

pinMode(L293N\_IN2, OUTPUT);

pinMode(L293N\_IN3, OUTPUT);

pinMode(L293N\_IN4, OUTPUT);

Serial.begin(9600);

BTSerial.begin(9600);

delay(1000);

}

void loop()

{

if (BTSerial.available()) {

if (BTConnect == false) {

BTConnect = true;

playBtConnectMelody();

}

// delay(100);

inString = "";

while (BTSerial.available()) {

inChar = BTSerial.read();

inString = inString + inChar;

}

Serial.println(inString);

if (inString == "#b=0#") {

robotStop();

}

else if (inString == "#b=9#" ||

inString == "#b=19#" ||

inString == "#b=29#" ||

inString == "#b=39#" ||

inString == "#b=49#") {

robotBreak();

}

else if (inString == "#b=1#") {

robotForward(200);

}

else if (inString == "#b=2#") {

robotReverse(200);

}

else if (inString == "#b=3#") {

robotTurnLeft(150);

}

else if (inString == "#b=4#") {

robotTurnRight(150);

}

else if (inString.startsWith("+DISC")) {

BTConnect = false;

delay(1000);

while (BTSerial.available()) {

BTSerial.read();

}

playBtDisconnectMelody();

}

}

}

void playMelody(int \*melody, int \*noteDurations, int notesLength)

{

pinMode(PIEZO, OUTPUT);

for (int thisNote = 0; thisNote < notesLength; thisNote++) {

int noteDuration = 1000 / noteDurations[thisNote];

tone(PIEZO, melody[thisNote], noteDuration);

int pauseBetweenNotes = noteDuration \* 1.30;

delay(pauseBetweenNotes);

noTone(PIEZO);

}

}

void robotStop()

{

digitalWrite(L293N\_ENA, LOW);

digitalWrite(L293N\_IN1, LOW);

digitalWrite(L293N\_IN2, LOW);

digitalWrite(L293N\_ENB, LOW);

digitalWrite(L293N\_IN3, LOW);

digitalWrite(L293N\_IN4, LOW);

}

void robotBreak()

{

digitalWrite(L293N\_ENA, HIGH);

digitalWrite(L293N\_IN1, HIGH);

digitalWrite(L293N\_IN2, HIGH);

digitalWrite(L293N\_ENB, HIGH);

digitalWrite(L293N\_IN3, HIGH);

digitalWrite(L293N\_IN4, HIGH);

}

void robotForward(int motorSpeed)

{

analogWrite(L293N\_ENA, motorSpeed);

digitalWrite(L293N\_IN1, LOW);

digitalWrite(L293N\_IN2, HIGH);

analogWrite(L293N\_ENB, motorSpeed);

digitalWrite(L293N\_IN3, HIGH);

digitalWrite(L293N\_IN4, LOW);

}

void robotReverse(int motorSpeed)

{

analogWrite(L293N\_ENA, motorSpeed);

digitalWrite(L293N\_IN1, HIGH);

digitalWrite(L293N\_IN2, LOW);

analogWrite(L293N\_ENB, motorSpeed);

digitalWrite(L293N\_IN3, LOW);

digitalWrite(L293N\_IN4, HIGH);

}

void robotTurnRight(int motorSpeed)

{

analogWrite(L293N\_ENA, motorSpeed);

digitalWrite(L293N\_IN1, HIGH);

digitalWrite(L293N\_IN2, LOW);

analogWrite(L293N\_ENB, motorSpeed);

digitalWrite(L293N\_IN3, HIGH);

digitalWrite(L293N\_IN4, LOW);

}

void robotTurnLeft(int motorSpeed)

{

analogWrite(L293N\_ENA, motorSpeed);

digitalWrite(L293N\_IN1, LOW);

digitalWrite(L293N\_IN2, HIGH);

analogWrite(L293N\_ENB, motorSpeed);

digitalWrite(L293N\_IN3, LOW);

digitalWrite(L293N\_IN4, HIGH);

}