#include <SoftwareSerial.h>

SoftwareSerial nodemcu(10,11);//tx,rx

String getValue(String, char, int);

long int data;

int firstVal, secondVal,thirdVal,fourthVal, fifthVal,sixthVal;

String myString; // complete message from arduino, which consistors of snesors data

char rdata; // received charactors

String cdata; // complete data

// for L298N motor driver pwm pins for speed control

int enA = 3;

int enB = 5;

int enC = 6;

int enD = 9;

//for digital direction control

// a b c d are motors 1,2 are switch to direction control

// a,b back tire

// c,d front tire

int ina1 = 2;

int ina2 = 4;

int inb1 = 7;

int inb2 = 8;

int inc1 = 12;

int inc2 = 13;

// arduino got no pins to support these

int ind1 = 0;

int ind2 = 1;

void setup()

{

Serial.begin(9600);

nodemcu.begin(9600);

pinMode(enA, OUTPUT);

pinMode(enB, OUTPUT);

pinMode(enC, OUTPUT);

pinMode(enD, OUTPUT);

pinMode(ina1, OUTPUT);

pinMode(ina2, OUTPUT);

pinMode(inb1, OUTPUT);

pinMode(inb2, OUTPUT);

pinMode(inc1, OUTPUT);

pinMode(inc2, OUTPUT);

pinMode(ind1, OUTPUT);

pinMode(ind2, OUTPUT);

analogWrite(enA, 0);

analogWrite(enB, 0);

analogWrite(enC, 0);

analogWrite(enD, 0);

}

void loop()

{

if(nodemcu.available() == 0 )

{

delay(100); // 100 milli seconds

}

if ( nodemcu.available() > 0 )

{

rdata = nodemcu.read();

myString = myString+ rdata;

Serial.print(rdata);

if( rdata == '\n')

{

Serial.println(myString);

// new code

firstVal = getValue(myString, ',', 0).toInt();//for left

secondVal = getValue(myString, ',', 1).toInt();//for right

thirdVal = getValue(myString, ',', 2).toInt();//for forward

fourthVal = getValue(myString, ',', 3).toInt();//for back

fifthVal = getValue(myString, ',', 4).toInt(); //speed for all

sixthVal = getValue(myString, ',', 5).toInt(); //for wheel drive switch

myString = "";

if(sixthVal=0)//2WD

{

if ( (firstVal == 0) && (secondVal == 0) && (thirdVal=0) && (fourthVal == 0)) // for normal position

{

analogWrite(enA,0);

analogWrite(enB,0);

analogWrite(enC,0);

analogWrite(enD,0);

digitalWrite(ina1,LOW);

digitalWrite(ina1,LOW);

digitalWrite(inb1,LOW);

digitalWrite(inb2,LOW);

digitalWrite(inc1,LOW);

digitalWrite(inc2,LOW);

digitalWrite(ind1,LOW);

digitalWrite(ind2,LOW);

}

else

{

if ( (thirdVal=1) && (fourthVal == 0)) // Forward direction

{

analogWrite(enA,fifthVal);

analogWrite(enB,fifthVal);

digitalWrite(ina1,HIGH);

digitalWrite(ina2, LOW);

digitalWrite(inb1,LOW);

digitalWrite(inb2,HIGH);

}

else if ( (thirdVal=0) && (fourthVal == 1)) // reverse direction

{

analogWrite(enA,fifthVal);

analogWrite(enB,fifthVal);

digitalWrite(ina1, LOW);

digitalWrite(ina2,HIGH);

digitalWrite(inb1,HIGH);

digitalWrite(inb2,LOW);

}

else //rest position

{

analogWrite(enA,0);

analogWrite(enB,0);

digitalWrite(ina1, LOW);

digitalWrite(ina2,LOW);

digitalWrite(inb1,LOW);

digitalWrite(inb2,LOW);

}

if ( (firstVal=1) && (secondVal == 0)) // left direction

{

analogWrite(enC,fifthVal);

analogWrite(enD,fifthVal);

digitalWrite(inc1,HIGH);

digitalWrite(inc2, LOW);

digitalWrite(ind1,LOW);

digitalWrite(ind2,LOW);

}

else if ( (firstVal=0) && (secondVal == 1)) // right direction

{

analogWrite(enC,fifthVal);

analogWrite(enD,fifthVal);

digitalWrite(inc2,LOW);

digitalWrite(inc1,LOW);

digitalWrite(ind2,LOW);

digitalWrite(ind1,HIGH);

}

else //rest position

{

analogWrite(enC,0);

analogWrite(enD,0);

digitalWrite(inc2,LOW);

digitalWrite(inc1,LOW);

digitalWrite(ind2,LOW);

digitalWrite(ind1,LOW);

}

}

}

if(sixthVal=1)// 4WD

{

if ( (firstVal == 0) && (secondVal == 0) && (thirdVal=0) && (fourthVal == 0)) // for normal position

{

analogWrite(enA,0);

analogWrite(enB,0);

analogWrite(enC,0);

analogWrite(enD,0);

digitalWrite(ina1,LOW);

digitalWrite(ina1,LOW);

digitalWrite(inb1,LOW);

digitalWrite(inb2,LOW);

digitalWrite(inc1,LOW);

digitalWrite(inc2,LOW);

digitalWrite(ind1,LOW);

digitalWrite(ind2,LOW);

}

else if ( (firstVal == 0) && (secondVal == 0) && (thirdVal=1) && (fourthVal == 0)) // Forward direction

{

analogWrite(enA,fifthVal);

analogWrite(enB,fifthVal);

analogWrite(enC,fifthVal);

analogWrite(enD,fifthVal);

digitalWrite(ina1,HIGH);

digitalWrite(ina2, LOW);

digitalWrite(inb1,LOW);

digitalWrite(inb2,HIGH);

digitalWrite(inc1,HIGH);

digitalWrite(inc2, LOW);

digitalWrite(ind1,LOW);

digitalWrite(ind2,HIGH);

}

else if ( (firstVal == 0) && (secondVal == 0) && (thirdVal=0) && (fourthVal == 1)) // Backward direction

{

analogWrite(enA,fifthVal);

analogWrite(enB,fifthVal);

analogWrite(enC,fifthVal);

analogWrite(enD,fifthVal);

digitalWrite(ina2,HIGH);

digitalWrite(ina1, LOW);

digitalWrite(inb2,LOW);

digitalWrite(inb1,HIGH);

digitalWrite(inc2,HIGH);

digitalWrite(inc1, LOW);

digitalWrite(ind2,LOW);

digitalWrite(ind1,HIGH);

}

else if ( (firstVal == 1) && (secondVal == 0) && (thirdVal=0) && (fourthVal == 0)) // Left direction

{

analogWrite(enA,fifthVal);

analogWrite(enB,fifthVal);

analogWrite(enC,fifthVal);

analogWrite(enD,fifthVal);

digitalWrite(ina1,HIGH);

digitalWrite(ina2, LOW);

digitalWrite(inb2,LOW);

digitalWrite(inb1,HIGH);

digitalWrite(inc1,HIGH);

digitalWrite(inc2, LOW);

digitalWrite(ind2,LOW);

digitalWrite(ind1,HIGH);

}

else if ( (firstVal == 0) && (secondVal == 1) && (thirdVal=0) && (fourthVal == 0)) // Right direction

{

analogWrite(enA,fifthVal);

analogWrite(enB,fifthVal);

analogWrite(enC,fifthVal);

analogWrite(enD,fifthVal);

digitalWrite(ina2,HIGH);

digitalWrite(ina1, LOW);

digitalWrite(inb1,LOW);

digitalWrite(inb2,HIGH);

digitalWrite(inc2,HIGH);

digitalWrite(inc1, LOW);

digitalWrite(ind1,LOW);

digitalWrite(ind2,HIGH);

}

}

}

}

}

String getValue(String data, char separator, int index)

{

int found = 0;

int strIndex[] = { 0, -1 };

int maxIndex = data.length() - 1;

for (int i = 0; i <= maxIndex && found <= index; i++) {

if (data.charAt(i) == separator || i == maxIndex) {

found++;

strIndex[0] = strIndex[1] + 1;

strIndex[1] = (i == maxIndex) ? i+1 : i;

}

}

return found > index ? data.substring(strIndex[0], strIndex[1]) : "";

}