Sumobot code book

The github repository can change breaking the links in the presentation. The code is here for simplicity.

The higher tree of the resources can be found here: <https://github.com/kaiwhata>

Contents

[Drive Forward Code 2](#_Toc501528464)

[Turning code 3](#_Toc501528465)

[Serial Control code 5](#_Toc501528466)

[Random exploration code (U/S sensor) 7](#_Toc501528467)

# Drive Forward Code

// Based on Sweep Example

// This example code is in the public domain.

#include <Servo.h>

Servo leftwheel; // create servo object to control a servo on a wheel

Servo rightwheel; // create another servo object to control the other wheel

void setup()

{

//assuming these are continuous rotation servos

leftwheel.attach(9); // attaches the servo on pin 9 to the servo object

rightwheel.attach(10); // attaches the servo on pin 10 to the servo object

}

void loop()

{

//for continuous rotation servos a number above 90 means turn in one direction

// and a number below 90 measn turn in the opposite direction.

// the further away from 90 the number is the faster the servo rotation

leftwheel.write(65); // tell servo to turn slowly in opposite directions

rightwheel.write(115); // this should make the robot drive roughly straight ahead

delay(500);

//drive\_forward(500, 5); //this uses the method below to achieve the same effect as the 3 lines of code above

leftwheel.write(90); // tell both servos to stop

rightwheel.write(90);

delay(500);

leftwheel.write(115); // tell servo to turn slowly in opposite directions

rightwheel.write(65); // this should make the robot drive roughly straight in reverse

delay(500);

leftwheel.write(90); // tell both servos to stop

rightwheel.write(90);

delay(500);

}

void drive\_forward(int drivetime, int speed)

{

//the speed parameter should be between 1 and 10

leftwheel.write(90-speed); // tell servo to turn slowly in opposite directions

rightwheel.write(90+speed); // this should make the robot drive roughly straight ahead

delay(drivetime);

}

# Turning code

// Based on Sweep Example

// This example code is in the public domain.

#include <Servo.h>

Servo leftwheel; // create servo object to control a servo on a wheel

Servo rightwheel; // create another servo object to control the other wheel

void setup()

{

//assuming these are continuous rotation servos

leftwheel.attach(9); // attaches the servo on pin 9 to the servo object

rightwheel.attach(10); // attaches the servo on pin 10 to the servo object

}

void loop()

{

//drive forward for 500 milliseconds at a speed of 5

drive\_forward(500, 25); //this uses the method below to achieve the same effect as the 3 lines of code above

//turn slowly left for 500 milliseconds

turn\_left(500);

//turn slowly right for 1 second

turn\_right(1000);

//

stop();

}

void drive\_forward(int drivetime, int speed)

{

//the speed parameter should be between 1 and 10

leftwheel.write(90-speed); // tell servo to turn slowly in opposite directions

rightwheel.write(90+speed); // this should make the robot drive roughly straight ahead

delay(drivetime);

}

void stop()

{

leftwheel.write(90); // stop left motor

rightwheel.write(90); // stop right motor

delay(100);

}

void turn\_left(int turntime)

{

leftwheel.write(90); // stops left wheel

rightwheel.write(115); // drives right wheel (this should make the robot drive roughly turn left)

delay(turntime);

}

void turn\_right(int turntime)

{

leftwheel.write(115); // drives left wheel

rightwheel.write(90); // stops right wheel (this should make the robot drive roughly turn right)

delay(turntime);

}

# Serial Control code

// Based on Sweep Example

// This example code is in the public domain.

#include <Servo.h>

Servo leftwheel; // create servo object to control a servo on a wheel

Servo rightwheel; // create another servo object to control the other wheel

void setup()

{

//assuming these are continuous rotation servos

leftwheel.attach(9); // attaches the servo on pin 9 to the servo object

rightwheel.attach(10); // attaches the servo on pin 10 to the servo object

//setup serial communication for debugging and testing threshold

Serial.begin(9600);

}

void loop()

{

//read data from serial connection

if (Serial.available() > 0) {

char inChar = Serial.read();

// print out the value you read:

Serial.println(inChar);

delay(1); // delay in between reads for stability

//if serial input is something

switch (inChar) {

case 'u':

// move robot forward

drive\_forward(500, 5);

break;

case 'd':

// move robot backward

drive\_backward(500, 5);

break;

case 'l':

// move mouse left

turn\_left(500);

break;

case 'r':

// move mouse right

turn\_right(500);

break;

}

stop();

}

}

void drive\_forward(int drivetime, int speed)

{

//the speed parameter should be between 1 and 10

leftwheel.write(90-speed); // tell servo to turn slowly in opposite directions

rightwheel.write(90+speed); // this should make the robot drive roughly straight ahead

delay(drivetime);

}

void drive\_backward(int drivetime, int speed)

{

//the speed parameter should be between 1 and 10

leftwheel.write(90+speed); // tell servo to turn slowly in opposite directions

rightwheel.write(90-speed); // this should make the robot drive roughly straight backward

delay(drivetime);

}

void stop()

{

leftwheel.write(90); // stop left motor

rightwheel.write(90); // stop right motor

delay(100);

}

void turn\_left(int turntime)

{

leftwheel.write(90); // stops left wheel

rightwheel.write(95); // drives right wheel (this should make the robot drive roughly turn left)

delay(turntime);

}

void turn\_right(int turntime)

{

leftwheel.write(95); // drives left wheel

rightwheel.write(90); // stops right wheel (this should make the robot drive roughly turn right)

delay(turntime);

}

# Random exploration code (U/S sensor)

/\*

HC-SR04 Ping distance sensor]

VCC to arduino 5v GND to arduino GND

Echo to Arduino pin 13 Trig to Arduino pin 12

to Arduino pin 11

to Arduino pin 10

\*/

#include <Servo.h>

#define trigPin 13

#define echoPin 12

#define led 11

#define led2 10

const int threshold = 10; //cm - sets how far robot responds

Servo leftwheel; // create servo object to control a servo on a wheel

Servo rightwheel; // create another servo object to control the other wheel

void setup() {

//setup serial communicstoin for debugging and testing threshold

Serial.begin (9600);

//setup ultrasouns sensor

pinMode(trigPin, OUTPUT);

pinMode(echoPin, INPUT);

//assuming these are continuous rotation servos

leftwheel.attach(9); // attaches the servo on pin 9 to the servo object

rightwheel.attach(10); // attaches the servo on pin 10 to the servo object

//setup random number generator - yes this could easily be improved!

randomSeed(analogRead(A0));

//randomSeed(2023);

}

void loop() {

long duration, distance;

digitalWrite(trigPin, LOW); //

delayMicroseconds(2); //

digitalWrite(trigPin, HIGH);

delayMicroseconds(10); //

digitalWrite(trigPin, LOW);

duration = pulseIn(echoPin, HIGH);

distance = (duration/2) / 29.1; //converts from 10 bit binary to cm

//controller logic

if (distance < threshold) {

int randomNumber = random(1,10);

Serial.print("Object detected: Turning ");

Serial.println(randomNumber\*100);

turn\_left(randomNumber\*100); //turns left for a random length of time

//turn randomly left or right?

}else{

Serial.println("Nothing Detected: Driving Forward");

//drive forward for one second

drive\_forward(1000, 15); //(time, speed)

}

/\* this code is not necessary for robot exploring

if (distance >= 200 || distance <= 0){

Serial.println("Out of range");

}else {

Serial.print(distance);

Serial.println(" cm");

}

\*/

Serial.print(distance);

Serial.println(" cm");

delay(500); // time between sensor checks

}

//method for controlling robot movement

void drive\_forward(int drivetime, int speed)

{

//the speed parameter should be between 1 and 10

leftwheel.write(90-speed); // tell servo to turn slowly in opposite directions

rightwheel.write(90+speed); // this should make the robot drive roughly straight ahead

delay(drivetime);

}

void stop()

{

leftwheel.write(90); // stop left motor

rightwheel.write(90); // stop right motor

delay(100);

}

void turn\_left(int turntime)

{

leftwheel.write(80); // stops left wheel

rightwheel.write(115); // drives right wheel (this should make the robot drive roughly turn left)

delay(turntime);

}

void turn\_right(int turntime)

{

leftwheel.write(115); // drives left wheel

rightwheel.write(80); // stops right wheel (this should make the robot drive roughly turn right)

delay(turntime);

}