//Demo to run a motor from H-Bridge motor controller

//connect motor controller pins to Arduino digital pins

int enA = 10;

int in1 = 9;

int in2 = 8;

int in3 = 7;

int in4 = 6;

void setup(){

// set all the motor control pins to outputs

pinMode(enA, OUTPUT);

pinMode(in1, OUTPUT);

pinMode(in2, OUTPUT);

pinMode(in3, OUTPUT);

pinMode(in4, OUTPUT);

}

void demoOne(){

// this function will run the motors in both directions at a fixed speed

// turn on motor A

digitalWrite(in1, HIGH);

digitalWrite(in2, LOW);

digitalWrite(in3, HIGH);

digitalWrite(in4, LOW);

// set speed to 200 out of possible range 0~255

analogWrite(enA, 200);

delay(2000);

// now change motor directions

digitalWrite(in1, LOW);

digitalWrite(in2, HIGH);

digitalWrite(in3, LOW);

digitalWrite(in4, HIGH);

delay(2000);

// now turn off motor

digitalWrite(in1, LOW);

digitalWrite(in2, LOW);

digitalWrite(in3, LOW);

digitalWrite(in4, LOW);

}

void loop(){

demoOne();

delay(1000);

}

//Relay control demonstration

int relay = 10;

int led = 8;

void setup() {

pinMode(relay, OUTPUT);

pinMode(led, OUTPUT);

digitalWrite(led, HIGH);

}

void loop() {

digitalWrite(relay, HIGH); //close relay contacts

delay(1000);

digitalWrite(relay, LOW); //open relay contacts

delay(1000);

}

//Real Time Clock Demonstration

#include <Wire.h>

#include "RTClib.h"

RTC\_DS1307 RTC;

void setup () {

Serial.begin(9600);

Wire.begin();

RTC.begin();

if (! RTC.isrunning()) {

Serial.println("RTC is NOT running!");

// following line sets the RTC to the date & time this sketch was compiled

RTC.adjust(DateTime(\_\_DATE\_\_, \_\_TIME\_\_));

}

}

void loop () {

DateTime now = RTC.now();

Serial.print(now.year(), DEC);

Serial.print('/');

Serial.print(now.month(), DEC);

Serial.print('/');

Serial.print(now.day(), DEC);

Serial.print(' ');

Serial.print(now.hour(), DEC);

Serial.print(':');

Serial.print(now.minute(), DEC);

Serial.print(':');

Serial.print(now.second(), DEC);

Serial.println();

delay(1000);

}

//Demonstration of turning a stepper motor 90 degrees in 1024 steps

#define IN1 8

#define IN2 9

#define IN3 10

#define IN4 11

int steps = 0;

boolean direction = true;

unsigned long last\_time;

unsigned long currentMillis ;

int steps\_left=1023; //4095 for 360 degrees

long time;

void setup(){

Serial.begin(115200);

pinMode(IN1, OUTPUT);

pinMode(IN2, OUTPUT);

pinMode(IN3, OUTPUT);

pinMode(IN4, OUTPUT);

}

void loop()

{

while(steps\_left>0){

currentMillis = micros();

if(currentMillis-last\_time>=1000){

stepper(1);

time=time+micros()-last\_time;

last\_time=micros();

steps\_left--;

}

}

Serial.println(time);

Serial.println("Wait...!");

delay(2000);

direction = !direction;

steps\_left=1023; //4095 for 360 degrees

}

void stepper(int xw){

for (int x=0;x<xw;x++){

switch(steps){

case 0:

digitalWrite(IN1, LOW);

digitalWrite(IN2, LOW);

digitalWrite(IN3, LOW);

digitalWrite(IN4, HIGH);

break;

case 1:

digitalWrite(IN1, LOW);

digitalWrite(IN2, LOW);

digitalWrite(IN3, HIGH);

digitalWrite(IN4, HIGH);

break;

case 2:

digitalWrite(IN1, LOW);

digitalWrite(IN2, LOW);

digitalWrite(IN3, HIGH);

digitalWrite(IN4, LOW);

break;

case 3:

digitalWrite(IN1, LOW);

digitalWrite(IN2, HIGH);

digitalWrite(IN3, HIGH);

digitalWrite(IN4, LOW);

break;

case 4:

digitalWrite(IN1, LOW);

digitalWrite(IN2, HIGH);

digitalWrite(IN3, LOW);

digitalWrite(IN4, LOW);

break;

case 5:

digitalWrite(IN1, HIGH);

digitalWrite(IN2, HIGH);

digitalWrite(IN3, LOW);

digitalWrite(IN4, LOW);

break;

case 6:

digitalWrite(IN1, HIGH);

digitalWrite(IN2, LOW);

digitalWrite(IN3, LOW);

digitalWrite(IN4, LOW);

break;

case 7:

digitalWrite(IN1, HIGH);

digitalWrite(IN2, LOW);

digitalWrite(IN3, LOW);

digitalWrite(IN4, HIGH);

break;

default:

digitalWrite(IN1, LOW);

digitalWrite(IN2, LOW);

digitalWrite(IN3, LOW);

digitalWrite(IN4, LOW);

break;

} //end of switch

setDirection();

} //end of for loop

} //end of stepper function

void setDirection(){

if(direction==1){

steps++;

}

if(direction==0){

steps--;

}

if(steps>7){

steps=0;

}

if(steps<0){

steps=7;

}

} //end of setDirection