#include <SPI.h>

int cs=10; //chip select

void setup() {

 Serial.begin(9600); //setup serial communication for feedback

 SPI.begin(); //let the SPI library handle mosi, miso, and ck.

 SPI.setBitOrder(MSBFIRST); //in accordance with datasheet for l6470. MSBFIRST

 pinMode(cs,OUTPUT);

 digitalWrite(cs, LOW); //pull cs low to send command

 SPI.transfer(0x00); //reset pos via byte command 1101 1000

 digitalWrite(cs, HIGH); //pull it high because datasheet says aftter every byte to pull high

 delay(1);

}

void loop(){

 digitalWrite(cs, LOW);

 SPI.transfer(0x00);

 digitalWrite(cs, HIGH);

 digitalWrite(cs, LOW);

 // SPI.transfer(0xD0); //get status command in hex //1101 0000

 digitalWrite(cs, HIGH);

 digitalWrite(cs, LOW);

 // SPI.transfer(0xD0); //get status command in hex //1101 0000

 digitalWrite(cs, HIGH);

 digitalWrite(cs, LOW);

 // SPI.transfer(0xD0); //get status command in hex //1101 0000

 digitalWrite(cs, HIGH);

 Serial.println(n); // output status return to see

 Serial.println(m);

 //set max\_speed register

 digitalWrite(cs, LOW);

 unsigned int maxSpeed = SPI.transfer(0x07); //initial command byte //0000 0111

 unsigned int maxSpeed1 = SPI.transfer(0x00); //MAx speed is a 10 bit register. This needs to be shifter twice to fill correct?

 unsigned int maxSpeed2 = SPI.transfer(0x00);

 digitalWrite(cs, HIGH); //Toggle back up

 Serial.print("Speed: ");

 Serial.println(maxSpeed);

 Serial.println(maxSpeed1);

 Serial.println(maxSpeed2);

 digitalWrite(cs, LOW);

 unsigned int elpos = SPI.transfer(0x02);

 unsigned int elpos1 = SPI.transfer(0x02);

 digitalWrite(cs, HIGH);

 Serial.print("elpos: ");

 Serial.println(elpos);

 Serial.println(elpos1);

 //test command

 digitalWrite(cs, LOW);

 SPI.transfer(0xD8); //reset pos 11011000

 digitalWrite(cs, HIGH);

 digitalWrite(cs, LOW);

 unsigned int speed = 0;

 SPI.transfer(0x14); //reset pos 0001 0100

 digitalWrite(cs, HIGH);

 //test command

 digitalWrite(cs, LOW);

 SPI.transfer(0x51); //Run command 0101 0001

 SPI.transfer(0x00);

 SPI.transfer(0x34); //0011 0100

 SPI.transfer(0xBC); //1011 1100

 digitalWrite(cs, HIGH);

 delay(500);

// HardHiz

 digitalWrite(cs, LOW);

 SPI.transfer(0xA8); //HIZ STOP 1010 1000

 digitalWrite(cs, HIGH);

 delay(500);

}