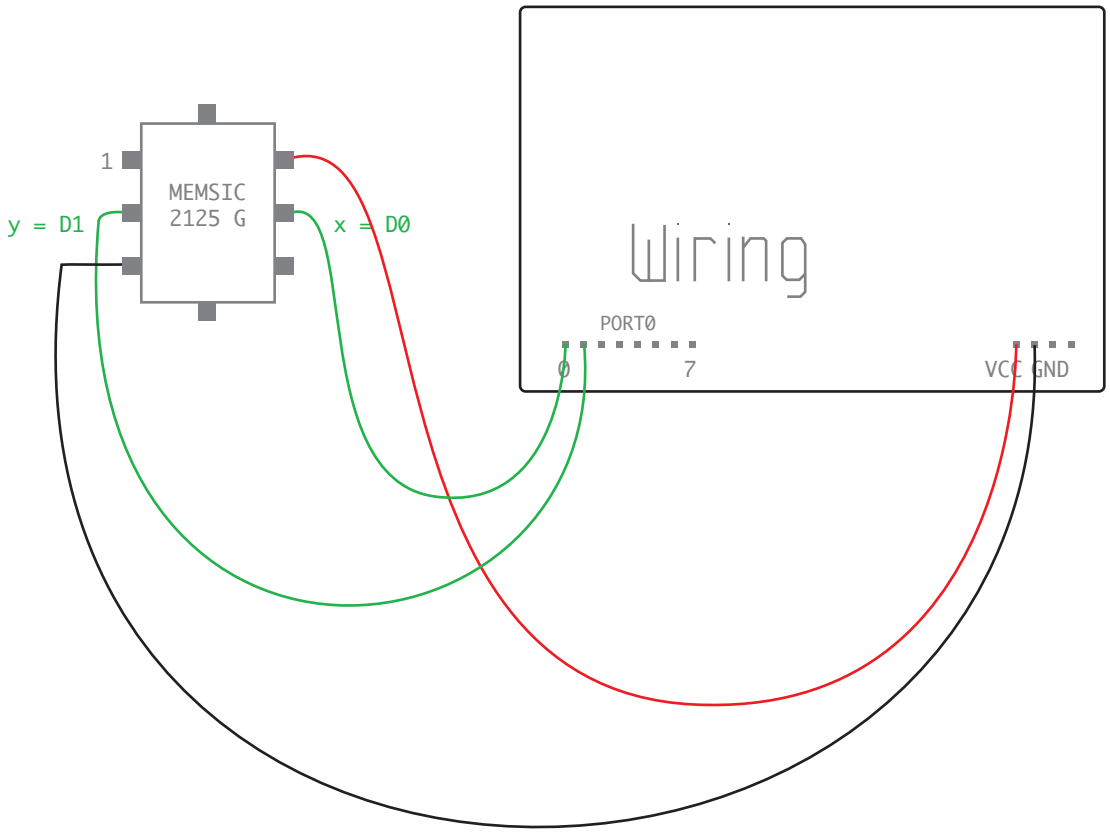
NEIGHBOURHOOD SATELLITE
x CIRCUITS x

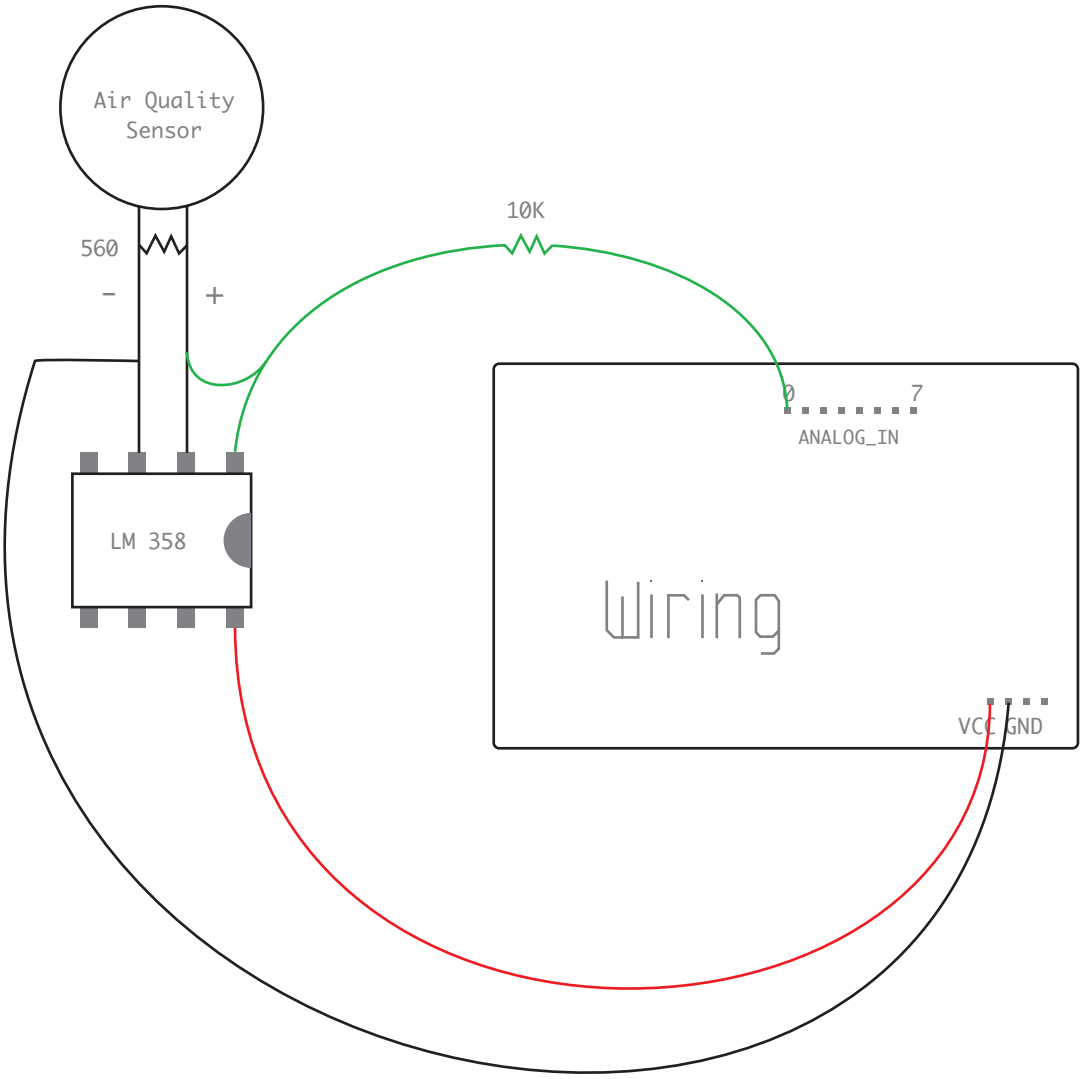
Here you find information on parts, circuits and Wiring code for:

- Accelerometer
- Air Pollution sensor
- Light sensor
- Motor

ACCELEROMETER

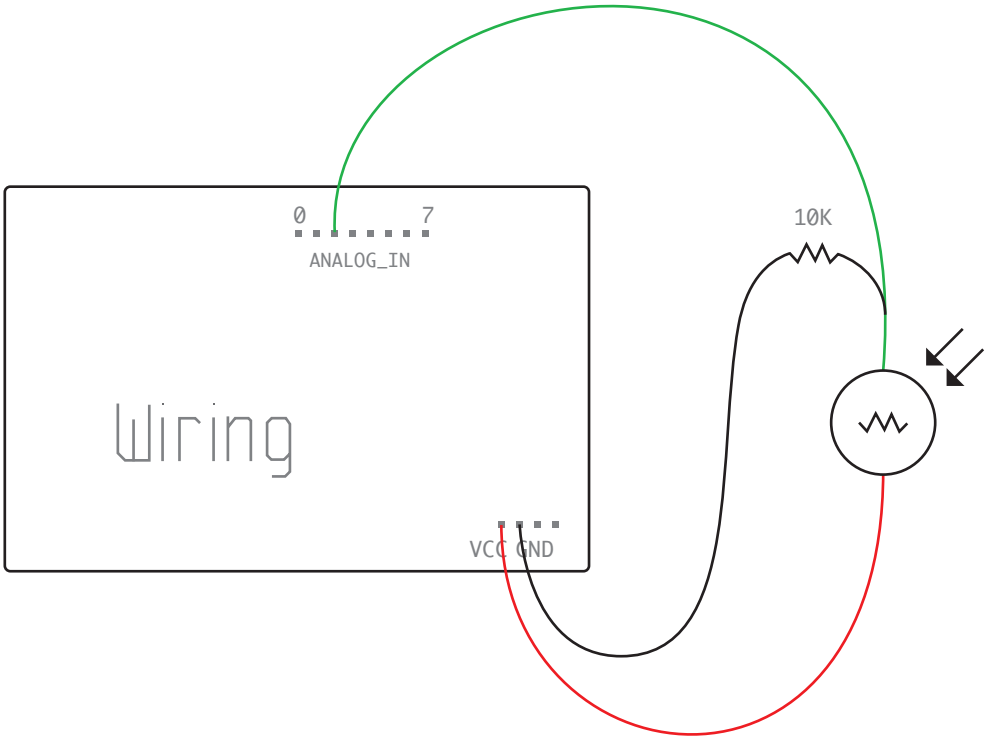


AIR QUALITY SENSOR



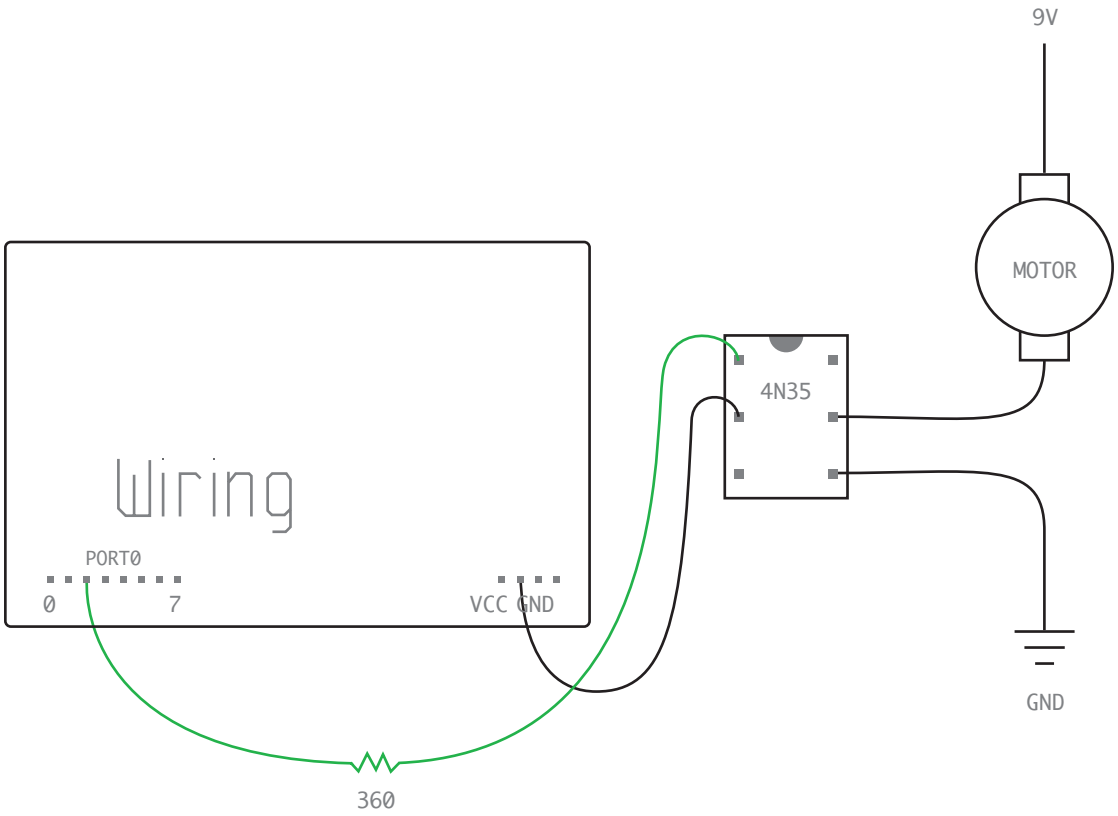
Parts:
- Air Quality Sensor: sensor and datasheet at dartsensors.com
- Resistors: 560/ 620 Ω, 10 kΩ
- Amplifier: LM358

LIGHT SENSOR



Parts:
- Light resistor
-10 kΩ resistor

MOTOR



Parts:
- a small DC motor
- Optocoupler 4N35
- 360Ω resistor
- 9V battery

WIRING CODE

by Nick Zambetti and Dave Mellis

```
/*
 * Pin Definitions
 */
byte pin_led = 48;
byte pin_x = 0;
byte pin_y = 1;
byte pin_motor = 2;
byte pin_lightsense = 2, pin_pollutionsense = 1;
int lastUpdate = 0;
byte pin_button1 = 24, pin_button2 = 25, pin_button3 = 27;
byte pin_buttonpower = 26;

void setPinModes(){
  pinMode(pin_led, OUTPUT);
  pinMode(pin_x, INPUT);
  pinMode(pin_y, INPUT);
  pinMode(pin_motor, OUTPUT);
}

/*
 * Global Variables
 */
int memsicDeadline = 0;
float memsicCounter = 0;
float memsicCounterX = 0;
float memsicCounterY = 0;
int memsicValueX = 0;
int memsicValueY = 0;
int motorCounter = 0;
byte memsicArraySize = 30;
byte memsicValuesX[] = { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
byte memsicValuesY[] = { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
byte memsicValuesIndex = -1;
byte memsicBalanceAdjustment = 0;
byte i = 0;
//int lastUpdate = 0;


/*
 * Wiring Mains
 */

void serialEvent(){
  motorCounter = serial;
}

void setup() {
  setPinModes();
  beginSerial(9600);
  printMode(SERIAL)
}
```

```
void loop() {
 while((LOW == digitalRead(pin_x)) && (LOW == digitalRead(pin_y))){
    continue;
  }
  memsicCounter = memsicCounterX = memsicCounterY = 0;
  memsicDeadline = millis() + 10;
  while(millis() < memsicDeadline){
    if(digitalRead(pin_x)){
      ++memsicCounterX;
    }
     if(digitalRead(pin_y)){
      ++memsicCounterY;
    }
    ++memsicCounter;
  }

  // sample x value and scale
  memsicValueX = (memsicCounterX/memsicCounter) * 1000;
  memsicValueY = (memsicCounterY/memsicCounter) * 1000;

  // clamp x values between 0 and 200
  memsicValueX = max(195, min(395, memsicValueX)) - 195; // experimentally
derived
  memsicValueY = max(220, min(420, memsicValueY)) - 220; // experimentally
derived

  // average x values
  memsicValuesIndex = (++memsicValuesIndex) % memsicArraySize;
  memsicValuesX[memsicValuesIndex] = memsicValueX;
  memsicValuesY[memsicValuesIndex] = memsicValueY;
  memsicValueX = 0;
  memsicValueY = 0;
  for(i = 0; i < memsicArraySize; ++i){
    memsicValueX += memsicValuesX[i];
    memsicValueY += memsicValuesY[i];
  }
  memsicValueX /= memsicArraySize;
  memsicValueY /= memsicArraySize;

  // compensate for what seems to be an off-kilter accelerometer
  memsicValueX = constrain(memsicValueX - memsicBalanceAdjustment, 0, 200);

  // if it's been more than 100 milliseconds
  if (millis() - lastUpdate > 100) {
    // report to serial listener
    print("%d,%d,%d,%d,%d,%d,%d\n", memsicValueX, memsicValueY,
      analogRead(pin_lightsense) / 4, analogRead(pin_pollutionsense)
    //serialWrite(0);
    lastUpdate = millis();
  }

  // operate motor
  if(1 < motorCounter){
    digitalWrite(pin_motor, HIGH);
    --motorCounter;
  }else{
    digitalWrite(pin_motor, LOW);
  }
}
```

LINKS

Air Quality Sensor from:
http://www.dart-sensors.com

Resources for Physical Computing:

http://wiring.org.co
http://www.potemkin.org
http://instantsoup.interaction-ivrea.it

http://www.neighbourhoodsatellites.com