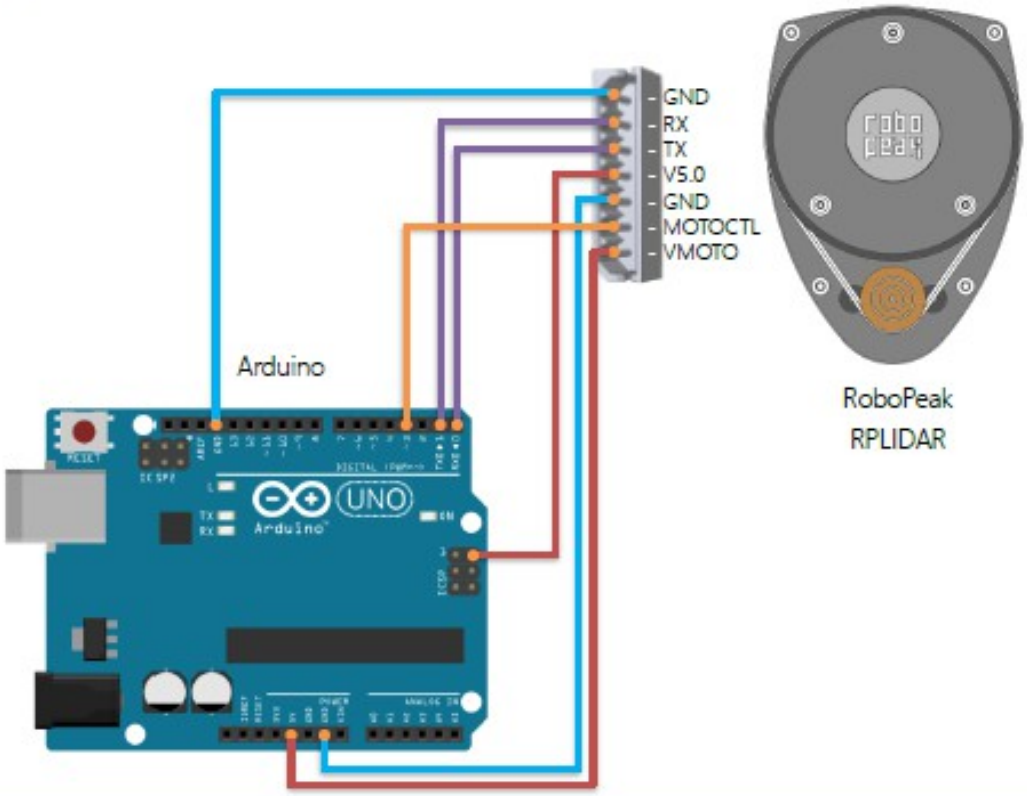
3. Example Demonstrations

Basics

- System Connection:

Connect RPLIDAR to your Arduino board as the following figure illustrates. The 2.5mm dupont wires can be used to make the connection.



RPLIDAR Pin	Arduino Pin	Description
GND	GND	Power Ground
V5.0	5V	5V Power Supply for the RPLIDAR Core
VMOTO	5V/ICSP Pin2	5V Power Supply for the RPLIDAR motor
RX	Pin1 (TXD)	Serial port, RX<-TX
TX	Pin0 (RXD)	Serial port, TX->RX
MOTOCTL	Pin3 (PWM)	RPLIDAR Motor speed control

If you just want the RPLIDAR rotation at maximum speed, the MOTOCTL pin can be connected to the 3V3 pin on the Arduino board.

```
/*
 * RoboPeak RPLIDAR Arduino Example
 * This example shows the easy and common way to fetch data from an RPLIDAR
 *
 * You may freely add your application code based on this template
 *
 * USAGE:
 * ---------------------------------
 * 1. Download this sketch code to your Arduino board
 * 2. Connect the RPLIDAR's serial port (RX/TX/GND) to your Arduino board (Pin 0 and Pin1)
 * 3. Connect the RPLIDAR's motor ctrl pin to the Arduino board pin 3
 */

/*
 * Copyright (c) 2014, RoboPeak
 * All rights reserved.
 * RoboPeak.com
 *
 * Redistribution and use in source and binary forms, with or without modification,
 * are permitted provided that the following conditions are met:
 *
 * 1. Redistributions of source code must retain the above copyright notice,
 *    this list of conditions and the following disclaimer.
 *
 * 2. Redistributions in binary form must reproduce the above copyright notice,
 *    this list of conditions and the following disclaimer in the documentation
 *    and/or other materials provided with the distribution.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY
 * EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
 * OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT
 * SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL,
 * SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT
 * OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR
 * TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE,
 * EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
 *
 */

// This sketch code is based on the RPLIDAR driver library provided by RoboPeak
#include <RPLidar.h>

// You need to create an driver instance
RPLidar lidar;
```

```
#define RPLIDAR_MOTOR 3 // The PWM pin for control the speed of RPLIDAR's motor.
// This pin should connected with the RPLIDAR's MOTOCTRL signal
//

#include <Adafruit_NeoPixel.h>
#define PIN 6
#define NUMPIXELS 24
Adafruit_NeoPixel pixels = Adafruit_NeoPixel(NUMPIXELS, PIN, NEO_GRB + NEO_KHZ800);
float SensorPixel[24];
float RedSensor[24], GreenSensor[24], BlueSensor[24];
float RedRange = 210.0 ;
float GreenRange = 1250.0;
float GreenSpan = GreenRange - RedRange;
float BlueRange = 3700.0;
float BlueSpan = BlueRange - GreenRange;
float Brightness = 64;
int delayval = 10;
int count = 0;
//
void setup() {
  // bind the RPLIDAR driver to the arduino hardware serial
  lidar.begin(Serial);

  // set pin modes
  pinMode(RPLIDAR_MOTOR, OUTPUT);
  //
  pixels.begin();
}

void loop() {
  if (IS_OK(lidar.waitPoint())) {
    float distance = lidar.getCurrentPoint().distance; //distance value in mm unit
    float angle    = lidar.getCurrentPoint().angle; //anglue value in degree
    bool  startBit = lidar.getCurrentPoint().startBit; //whether this point is belong to a new scan
    byte  quality  = lidar.getCurrentPoint().quality; //quality of the current measurement
    //perform data processing here...
    count++;
    if (angle > 352.5 && angle <= 360 || angle >= 0 && angle <= 7.5) {   //   0      pixel  0
      SensorPixel[0] = distance;
    }
    if (angle >   7.5 && angle <  22.5) {        //  15    pixel  1
      SensorPixel[1] = distance;
    }
    if (angle >  22.5 && angle <  37.5) {        //  30    pixel  2
      SensorPixel[2] = distance;
    }
    if (angle >  37.5 && angle <  52.5) {        //  45    pixel  3
      SensorPixel[3] = distance;
    }
```

```
if (angle >  52.5 && angle <  67.5) {       //  60   pixel  4
  SensorPixel[4] = distance;
}
if (angle >  67.5 && angle <  82.5) {       //  75   pixel  5
  SensorPixel[5] = distance;
}
if (angle >  82.5 && angle <  97.5) {       //  90   pixel  6
  SensorPixel[6] = distance;
}
if (angle >  97.5 && angle < 112.5) {       // 105   pixel  7
  SensorPixel[7] = distance;
}
if (angle > 112.5 && angle < 127.5) {       // 120   pixel  8
  SensorPixel[8] = distance;
}
if (angle > 127.5 && angle < 142.5) {       // 135   pixel  9
  SensorPixel[9] = distance;
}
if (angle > 142.5 && angle < 157.5) {       // 150   pixel 10
  SensorPixel[10] = distance;
}
if (angle > 157.5 && angle < 172.5) {       // 165   pixel 11
  SensorPixel[11] = distance;
}
if (angle > 172.5 && angle < 187.5) {       // 180   pixel 12
  SensorPixel[12] = distance;
}
if (angle > 187.5 && angle < 202.5) {       // 195   pixel 13
  SensorPixel[13] = distance;
}
if (angle > 202.5 && angle < 217.5) {       // 210   pixel 14
  SensorPixel[14] = distance;
}
if (angle > 217.5 && angle < 232.5) {       // 225   pixel 15
  SensorPixel[15] = distance;
}
if (angle > 232.5 && angle < 247.5) {       // 240   pixel 16
  SensorPixel[16] = distance;
}
if (angle > 247.5 && angle < 262.5) {       // 255   pixel 17
  SensorPixel[17] = distance;
}
if (angle > 262.5 && angle < 277.5) {       // 270   pixel 18
  SensorPixel[18] = distance;
}
if (angle > 277.5 && angle < 292.5) {       // 285   pixel 19
  SensorPixel[19] = distance;
}
if (angle > 292.5 && angle < 307.5) {       // 300   pixel 20
```

```
    SensorPixel[20] = distance;
   }
   if (angle > 307.5 && angle < 322.5) {      // 315   pixel 21
    SensorPixel[21] = distance;
   }
   if (angle > 322.5 && angle < 337.5) {      // 330   pixel 22
    SensorPixel[22] = distance;
   }
   if (angle > 337.5 && angle < 352.5) {      // 345   pixel 23
    SensorPixel[23] = distance;
   }
 }

 else {
  analogWrite(RPLIDAR_MOTOR, 0); //stop the rplidar motor

  // try to detect RPLIDAR...
  rplidar_response_device_info_t info;
  if (IS_OK(lidar.getDeviceInfo(info, 100))) {
    // detected...
    lidar.startScan();

    // start motor rotating at max allowed speed
    analogWrite(RPLIDAR_MOTOR, 255);
    delay(1000);
  }
 }
 if (count >= 360) {
  count = 0;
         //
  for (int i = 0; i <= 23; i++) {
    if (SensorPixel[i] <= RedRange)
      RedSensor[i] = SensorPixel[i] * (Brightness / RedRange), GreenSensor[i] = 0, BlueSensor[i] = 0;
    else if (SensorPixel[i] > RedRange && SensorPixel[i] <= GreenRange)
      GreenSensor[i] = SensorPixel[i] * (Brightness / GreenSpan), RedSensor[i] = 0, BlueSensor[i] = 0;
    else if (SensorPixel[i] > GreenRange && SensorPixel[i] <= BlueRange)
      BlueSensor[i] = SensorPixel[i] * (Brightness / BlueSpan), RedSensor[i] = 0, GreenSensor[i] = 0;
    else if (SensorPixel[i] > BlueRange)
      RedSensor[i] = Brightness, GreenSensor[i] = Brightness, BlueSensor[i] = Brightness;

    pixels.setPixelColor(i, pixels.Color(RedSensor[i], GreenSensor[i], BlueSensor[i]));
  }

  pixels.show();
 }
}
```

Parts List:

1) Arduino Uno Rev 3
2) RoboPeak LIDAR
3) Adafruit NeoPixel 24 Ring
4) SparkFun Breadboard Power Supply 5V/3.3V PRT-00114
5) 1000uF 35V capacitor
6) 470ohm 1/4watt resistor
7) jumper wires
8) 12V AA Battery Tray with Plug End (8-cell) Part: SC-6049-NJP
9) y power cable (1) female receptacle (2) male plugs – one for breadboard power supply and one for Arduino
10) Global Specialties Proto-Board 104

I have the basics out of the way. Next I want to work out navigation for a robot.