```
void loop() {
  int value = 0;

  if(Serial.available()){
    Serial.println("New values received");
    for(int i=0; i<5; i++) {
      int value = Serial.read();
      Serial.print("Value: ");
      Serial.println(value / 100.0);
      curr_color[i] = toColor(value / 100.0);
    }
    update_leds = true;
  };

  if (digitalRead(BUTTON_PIN) == LOW) {
    update_leds = true;
    delay(200);
    buttonStuffs();
    Serial.println("Button pressed");
  }

  if (update_leds) {
    Serial.println("Updating LEDs");
    for(int i=12; i<NUMPIXELS; i++) pixels.setPixelColor(i,
curr_color[curr_selection*1]);
    if (curr_selection == -2) {
      for(int i=12; i<NUMPIXELS; i++) pixels.setPixelColor(i, 0);
    }
    pixels.show();
    update_leds = false;
  }
}
```

New values received

Changed measure

Update LEDs

```
void buttonStuffs() {
  int shift = 0;
  if (curr_selection >= 0) pixels.setPixelColor(curr_selection * 3 +
  shift, 0);
  curr_selection++;
  if (curr_selection == 4) {
    curr_selection = -2;
  }
  if (curr_selection >= 0) pixels.setPixelColor(curr_selection * 3 +
  shift,
  pixels.Color(0,0,0.1*255));
}

uint32_t toColor(float value)
 {
  int red, green, blue;
  blue = 0;
  if (value < 0.5) {
    red = 255;
    green = (int) 255*2*value;
  }
  else {
    red = (int) 255 - 255*(value-0.5)*2;
    green = 255;
  }

  char stringos[50];
  sprintf(stringos, "Color: (%d, %d, %d)\n", red, green, blue);
  Serial.println(stringos);
  return pixels.Color(0.1*red,0.1*green,0.1*blue);
}
```

Change selection

Convert to color