|  |
| --- |
| 거리측정  #include <Servo.h>  Servo servo;  int servoPin = 9;  int angle = 0; // servo position in degrees  int mMax = -1;  void setup()  {  Serial.begin(9600);  pinMode(2,OUTPUT); // 센서 Trig 핀  pinMode(3,INPUT); // 센서 Echo 핀  servo.attach(servoPin);  }  void loop()  {  long duration;    digitalWrite(2,HIGH); // 센서에 Trig 신호 입력  delayMicroseconds(10); // 10us 정도 유지  digitalWrite(2,LOW); // Trig 신호 off  duration = pulseIn(3,HIGH); // Echo pin: HIGH->Low 간격을 측정  float cm = microsecondsToCentimeters(duration); // 거리(cm)로 변환  if(mMax<0){  mMax = cm;  }else{  mMax = max(mMax, cm);  }  float mPer = (cm / mMax) \* 100.0;  int sAngle = mPer \* 180 / 100;  Serial.print(cm);  Serial.print(" cm, max:");  Serial.print(mMax);  Serial.print(", percent:");  Serial.print(mPer);  Serial.print(", setAngle:");  Serial.print(sAngle);  Serial.println();  servo.write(sAngle);  delay(150);  }  long microsecondsToInches(long microseconds)  {  // According to Parallax's datasheet for the PING))), there are  // 73.746 microseconds per inch (i.e. sound travels at 1130 feet per  // second). This gives the distance travelled by the ping, outbound  // and return, so we divide by 2 to get the distance of the obstacle.  // See: http://www.parallax.com/dl/docs/prod/acc/28015-PING-v1.3.pdf  return microseconds / 74 / 2;  }    long microsecondsToCentimeters(long microseconds)  {  // The speed of sound is 340 m/s or 29 microseconds per centimeter.  // The ping travels out and back, so to find the distance of the  // object we take half of the distance travelled.  return microseconds / 29 / 2;  } |

|  |
| --- |
| 릴레이  #include <Servo.h>  void setup()  {  Serial.begin(9600);  pinMode(13,OUTPUT); // 센서 Echo 핀  }  void loop()  {  digitalWrite(13,HIGH); // Trig 신호 off  delay(2000);  digitalWrite(13,LOW);  delay(2000);  }  long microsecondsToInches(long microseconds)  {  // According to Parallax's datasheet for the PING))), there are  // 73.746 microseconds per inch (i.e. sound travels at 1130 feet per  // second). This gives the distance travelled by the ping, outbound  // and return, so we divide by 2 to get the distance of the obstacle.  // See: http://www.parallax.com/dl/docs/prod/acc/28015-PING-v1.3.pdf  return microseconds / 74 / 2;  }    long microsecondsToCentimeters(long microseconds)  {  // The speed of sound is 340 m/s or 29 microseconds per centimeter.  // The ping travels out and back, so to find the distance of the  // object we take half of the distance travelled.  return microseconds / 29 / 2;  } |

|  |
| --- |
| 거리측정 + 릴레이  #include <Servo.h>  void setup()  {  Serial.begin(9600);  pinMode(13,OUTPUT); // 센서 Echo 핀    pinMode(2,OUTPUT); // 센서 Trig 핀  pinMode(3,INPUT); // 센서 Echo 핀    }  void loop()  {  //sound  long duration;  digitalWrite(2,HIGH); // 센서에 Trig 신호 입력  delayMicroseconds(10); // 10us 정도 유지  digitalWrite(2,LOW); // Trig 신호 off  duration = pulseIn(3,HIGH); // Echo pin: HIGH->Low 간격을 측정  float cm = microsecondsToCentimeters(duration); // 거리(cm)로 변환  Serial.print(cm);  Serial.println(" cm");    digitalWrite(13,HIGH); // Trig 신호 off  delay(10);  if(cm<15.0){  digitalWrite(13,LOW);  delay(3000);  digitalWrite(13,HIGH);  delay(3000);  }  //digitalWrite(13,LOW);  // delay(2000);  }  long microsecondsToInches(long microseconds)  {  // According to Parallax's datasheet for the PING))), there are  // 73.746 microseconds per inch (i.e. sound travels at 1130 feet per  // second). This gives the distance travelled by the ping, outbound  // and return, so we divide by 2 to get the distance of the obstacle.  // See: http://www.parallax.com/dl/docs/prod/acc/28015-PING-v1.3.pdf  return microseconds / 74 / 2;  }    long microsecondsToCentimeters(long microseconds)  {  return microseconds / 29 / 2;  } |

|  |
| --- |
| 마그네틱  #include <Servo.h>  int S = A0; // INPUT PIN  void setup()  {  Serial.begin(9600);  pinMode(A1,INPUT); // 센서 Echo 핀  }  void loop()  {  Serial.print("S=");  Serial.println(analogRead(S), DEC);  delay(100);  }  long microsecondsToInches(long microseconds)  {    return microseconds / 74 / 2;  }    long microsecondsToCentimeters(long microseconds)  {    return microseconds / 29 / 2;  } |