**청소로봇 개발보고서**

**팀원:** 전채린

|  |
| --- |
| **1. 개발목표와 특징** |

*(각 팀에서 개발하고자 하는 제안시스템의 특징을 반영한 개발목표를 기술)*

|  |
| --- |
| 스스로 청소해주는 로봇을 만들어 가정에서 보다 편리하게 생활할 수 있게 도와주는 것이 목표이다. 방 모양과 가구 배치를 인식하고 맵핑할 수 있는 청소로봇을 구현하고 싶다.  장애물감지, 추락방지, 충돌방지, 바닥감지, 빛 감지, 전면감지, 먼지감지, 레이저거리측정, 각도, 접촉, 모서리감지 등 다양한 센서들을 활용해 청소로봇을 구현하고 싶다. 그뿐만 아니라 공유기능을 통해 가족들과 함께 청소로봇을 사용할 수 있는 공유 기능, 시간, 요일, 구역을 설정해 원하는 시간대에 청소를 예약할 수 있는 청소기능, 화장실, 신발장, 애견패드 등 원하지 않는 부분을 금지구역 또는 가상벽을 설정해 청소영역을 제한할 수 있는 금지구역과 가상벽 설정 기능, 원하는 구역을 지정해 집중적으로 청소 가능한 지정구역 설정 기능, 앱만으로 마치 리모컨처럼 별도의 리모컨 없이도 앱에서 간단한 리모컨 조작이 가능한 편리한 리모컨 기능, 청소를 원하지 않는 시간대를 설정할 수 있는 청소 금지 시간 설정 기능, 청소 시작부터 완료, 충전 복귀까지 음성안내를 통해 실시간으로 청소상황과 제품 상태를 파악할 수 있는 음성 안내 기능을 추가해 사물인터넷(IOT)으로 확대해 보고 싶다. |

|  |
| --- |
| **2. 개발필요성** |

*(왜 본 시스템 개발이 필요한지 기술적 환경 및 수요 배경을 중심으로 기술)*

|  |
| --- |
| 청소로봇은 집안을 마음껏 돌아다닐 수 있고 이미 형성된 시장을 활용해 가정용 로봇 확산을 꾀할 수 있다. 또한, 로봇 스스로가 눈으로 확보한 데이터를 인터넷을 통해 클라우드에 연동하고 데이터를 스마트홈 전반에 활용할 수 있다.  청소로봇이 확보한 데이터 소스를 활용해 가정 내 스마트홈 기기와의 연동을 강화하면 집안 곳곳의 환경이나 구조에 대한 이해도가 높아지고, 센서에 카메라를 더하게 되면 가정 내 정보를 시각화할 수 있어 활용도는 더 높아질 것이다. 2016년 기준으로 PC나 스마트폰, 웨어러블 기기 등을 통틀어 1인이 생성하는 데이터는 650MB가량이라고 한다. 거기에 더해 자율주행차가 하루 동안 만들어낼 데이터는 무려 4TB에 달할 것으로 예상된다.  사물인터넷(IOT)화 역시 가정에 있는 모든 것을 인터넷에 연결하고 상호 연동하는 스마트홈으로 단순 청소용 로봇이 아닌 가정용 로봇과 심부름 로봇으로 볼 수 있다. 심부름 로봇에 관성 센서를 내장한 카메라를 탑재하면, 집안 상황을 인식하고 스스로 계획을 세워 가사 작업을 할 수 있을 것이다. 주위 환경을 인식할 수 있는 카메라를 단 로봇청소기가 실용화된다면 방 안에 있는 문이나 의자 다리 위치까지 파악할 수 있고, 집안일을 계획하고 작업을 수행하는 복잡한 동작도 가능할 것이다. 또한, 인간과 로봇이 의사소통할 수 있는 시스템을 구축해 음성 명령을 내리면 로봇이 이를 이해하고 실행할 수 있을 것이다.  그뿐만 아니라 클라우드에 대용량 데이터를 저장해두고 로봇이 이 데이터를 이용해 가사 작업을 학습할 수 있게 할 수 있다. 와이파이를 통해 스마트폰과 연동, 원격 조작을 할 수 있고 카메라를 갖춰 가구 배치나 방 형태를 파악해 직접 지도를 생성할 수 있다. 또한, 한 번 청소한 장소를 여러 번 청소하지 않는 등 효율성을 높일 수 있다. 센서와 시각화 시스템을 갖춰 가구 배치와 방 형태를 이해할 수 있는 지도를 만들기 때문에 청소로봇이 스스로 지금 어디에 있는지 또 다음에 어디로 가야 할지 파악할 수 있다. 따라서 직접 청소하지 않아도 청소 로봇이 집안 곳곳을 청소해주기 때문에 가정에서 보다 편리하게 생활할 수 있다. |

|  |
| --- |
| **3. H/W 설계** |

*H/W를 어떻게 구현했는지를 그림, 사진 등으로 설명, 사용 센서, 부품 들을 나열.*

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 아두이노비 BOE 쉴드, USB 케이블, LED, 저항, 점프 와이어, 서보모터, 수동부저, 적외선 수신기, 적외선 LED, 포토트랜지스터, 초음파 센서, I2C LCD 디스플레이  1) 아두이노비 BOE 쉴드    2) 서버모터     |  |  | | --- | --- | | Servo | 아두이노비 | | 왼쪽 | D12 | | 오른쪽 | D13 |   3) 적외선 수신기 및 적외선 LED    4) 포토트랜지스터     |  |  | | --- | --- | | Phototransistor | 아두이노비 | | + | 5V | | - | GND |   5) 초음파 센서     |  |  | | --- | --- | | HC-SR04 | 아두이노비 | | VCC | 5V | | Trig | D13 | | Echo | D12 | | GND | GND |   6) I2C LCD 디스플레이     |  |  | | --- | --- | | I2C LCD | 아두이노비 | | SCL | SCL | | SDA | SDA | | VCC | 5V | | GND | GND |   7) 수동부저     |  |  | | --- | --- | | Buzzer | 아두이노비 | | + | D4 | | - | GND |   전자기기이(가) 표시된 사진  자동 생성된 설명  텍스트, 전자기기이(가) 표시된 사진  자동 생성된 설명    전자기기이(가) 표시된 사진  자동 생성된 설명 |
| **4. S/W 설계** |

*S/W를 어떻게 구현할 지를 설명, 스케치 코드 첨부*

|  |
| --- |
| 적외선 수신기와 적외선 LED를 통해 전방의 장애물을 감지하고, 바닥, 전면, 접촉, 모서리를 감지하고, 이를 통해 추락과 충돌을 방지하고 모터의 움직임을 제어한다.  초음파 센서를 통해 후방의 장애물을 감지하고, 이를 통해 모터의 움직임을 제어한다.  포토트랜지스터를 통해 빛을 감지하고, 이를 통해 LED의 밝기와 모터의 움직임을 제어한다.  LCD 화면을 통해 모터의 움직임을 출력한다.  전방 왼쪽과 오른쪽에서 장애물이 감지되면, 20ms 후진한다.  전방 왼쪽에서 장애물이 감지되면, 20ms 우회전한다.  전방 오른쪽에서 장애물이 감지되면, 20ms 좌회전한다.  후방에서 장애물이 10cm보다 가까워지면, 20ms 전진한다. |

|  |
| --- |
| **5. 구현 소감** |

*개발 과정에 있었던 일과 느낌을 소개*

|  |
| --- |
| 아두이노에 점프와이어, 저항 등이 많이 연결되서 주행할 때 선들이 서로 닿으면서 혼선이 발생하는 어려움이 있었다. 또한, 시연 장소가 협소하고 장애물이 많아 아두이노가 작동하는 데 어려움이 있었다. |

|  |
| --- |
| **6. 향후 보완 사항** |

*향후 보완이 필요하다고 생각되는 사항 기술*

|  |
| --- |
| 향후 방 모양과 가구 배치를 인식하고 맵핑(Mapping)할 수 있는 청소로봇을 구현하고 싶다. 공유기능을 통해 가족들과 함께 청소로봇을 사용할 수 있는 공유 기능(Sharing), 시간, 요일, 구역을 설정해 원하는 시간대에 청소를 예약할 수 있는 청소기능(Cleaning), 화장실, 신발장, 애견패드 등 원하지 않는 부분을 금지구역 또는 가상벽을 설정해 청소영역을 제한할 수 있는 금지구역과 가상벽 설정 기능(Virtual wall), 원하는 구역을 지정해 집중적으로 청소 가능한 지정구역 설정 기능(Point), 앱만으로 마치 리모컨처럼 별도의 리모컨 없이도 앱에서 간단한 리모컨 조작이 가능한 편리한 리모컨 기능(Control), 청소를 원하지 않는 시간대를 설정할 수 있는 청소 금지 시간 설정 기능(Time), 청소 시작부터 완료, 충전 복귀까지 음성안내를 통해 실시간으로 청소상황과 제품 상태를 파악할 수 있는 음성 안내 기능(Voice)을 추가해 사물인터넷(IOT)으로 확대해 보고 싶다. 이에 더해 클라우드에 대용량 데이터를 저장해두고 청소로봇이 이 데이터를 이용해 가사 작업을 학습할 수 있게 발전시키고 싶다. |

|  |
| --- |
| **7. 참고 문헌** |

*개발 과정에 참고한 도서, 사이트 등을 기술*

|  |
| --- |
| IOT프로그래밍 강의 자료 |

**첨부1. 스케치 소스 코드**

/\* IOT프로그래밍 기말프로젝트 청소로봇 스케치코드 컴퓨터소프트웨어공학과 2-YA 20202296 전채린 \*/

/\* CleanUpRobot \*/

#include <Servo.h> // Include servo library

#include <Wire.h> // Include wire library

#include <LiquidCrystal\_I2C.h> // lcd 1602 i2c library

Servo servoLeft; // Declare left and right servos

Servo servoRight;

#define trigPin 13 // Declare trigPin 13 and echoPin 12

#define echoPin 12

LiquidCrystal\_I2C lcd(0x27, 16, 2);

int irDetect(int irLedPin, int irReceiverPin, long frequency);

void maneuver(int speedLeft, int speedRight, int msTime);

long rcTime(int pin);

float volts(int adPin);

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;

}

void setup() { // Built-in initialization block

pinMode(10, INPUT); pinMode(9, OUTPUT);//Left IR LED & Receiver

pinMode(3, INPUT); pinMode(2, OUTPUT);//Right IR LED & Receiver

pinMode(8, OUTPUT); pinMode(7, OUTPUT);// Indicator LEDs

pinMode(trigPin, OUTPUT); // trigPin OUTPUT

pinMode(echoPin, INPUT); // echoPin INPUT

tone(4, 3000, 1000); // Play tone for 1 second

delay(1000); // Delay to finish tone

servoLeft.attach(13); // Attach left signal to pin 13

servoRight.attach(12); // Attach right signal to pin 12

lcd.begin(); // lcd start

lcd.backlight(); // backlight on

lcd.noCursor(); // no cursor

lcd.noBlink(); // no blink cursor

}

void loop() { // Main loop auto-repeats

long cm, duration;

digitalWrite(trigPin, LOW);

delayMicroseconds(2);

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

duration = pulseIn(echoPin, HIGH);

// convert the time into a distance

cm = microsecondsToCentimeters(duration);

int irLeft = irDetect(9, 10, 38000); // Check for object on left

int irRight = irDetect(2, 3, 38000);// Check for object on right

digitalWrite(8, !irLeft); // LED states opposite of IR

digitalWrite(7, !irRight);

if((irLeft == 0) && (irRight == 0)) { // If both sides detect

maneuver(-100, -100, 20); // Backward 20 milliseconds

lcd.clear();

lcd.print("Backward");

if(cm < 10) {

maneuver(100, 100, 20); // Forward 20 ms

lcd.clear();

lcd.print("Forward");

}

}

else if(irLeft == 0) { // If only left side detects

maneuver(100, -100, 20); // Right for 20 ms

lcd.clear();

lcd.print("Right");

}

else if(irRight == 0) { // If only right side detects

maneuver(-100, 100, 20); // Left for 20 ms

lcd.clear();

lcd.print("Left");

}

else { // Otherwise, no IR detects

maneuver(100, 100, 20); // Forward 20 ms

lcd.clear();

lcd.print("Forward");

}

if(volts(A3) > 3.5) // If A3 voltage greater than 3.5

{

servoLeft.detach(); // Stop servo signals

servoRight.detach();

lcd.clear();

lcd.print("Stop");

}

delay(50); // delay 50ms

}

int irDetect(int irLedPin, int irReceiverPin, long frequency)

{

tone(irLedPin, frequency, 8); // IRLED 38 kHz for at least 1 ms

delay(1); // Wait 1 ms

int ir = digitalRead(irReceiverPin); // IR receiver -> ir variable

delay(1); // Down time before recheck

return ir; // Return 1 no detect, 0 detect

}

void maneuver(int speedLeft, int speedRight, int msTime)

{ // speedLeft, speedRight ranges: Backward Linear

// Stop Linear Forward

// -100......0......100

// Set Left servo speed

servoLeft.writeMicroseconds(1500 + speedLeft);

// Set right servo speed

servoRight.writeMicroseconds(1500 - speedRight);

if(msTime==-1) // if msTime = -1

{

servoLeft.detach(); // Stop servo signals

servoRight.detach();

}

delay(msTime); // Delay for msTime

}

long rcTime(int pin) // returns decay time

{

pinMode(pin, OUTPUT); // Charge capacitor

digitalWrite(pin, HIGH); // by setting pin output-high

delay(5); // for 5 ms

pinMode(pin, INPUT); // Set pin to input

digitalWrite(pin, LOW); // with no pullup

long time = micros(); // Mark the time

while(digitalRead(pin)); // Wait for voltage < threshold

time = micros() - time; // Calculate decay time

return time; // Return decay time

}

float volts(int adPin) //Measures volts at adPin, Returns floating point voltage

{

return float(analogRead(adPin)) \* 5.0 / 1023.0;

}