<별표 3:설계모델 명세서>

|  |  |  |  |
| --- | --- | --- | --- |
| 문서분류 | 설계모델 명세서 | 과제번호 | SWD - 003 |
|  | | | |
|  | | | |
| **IoT를 활용한 반려동물 케어 장치** | | | |
|  | | | |
| |  |  | | --- | --- | | **과목명** | **종합설계프로젝트** | | **교수명** | **이 인 복 교수님** | | **학과** | **소프트웨어학과** | | **학번** | **2013121196 장준혁**  **2015124188 장우진**  **2016125086 김유진** | | **제출일** | **2019.12.26** | | | | |
|  | | | |

<별표 3-1>

|  |  |  |  |
| --- | --- | --- | --- |
| **설계모델 명세서 : 개념설계** | | **날 짜** | **2019.12.21** |
| **작성자** | **김유진, 장우진** |
| 프로젝트명 | 비컨 기기를 활용한 반려동물 케어 IoT 장치 | | |
| 프로젝트 기간 | 2019.09.02. - 2019.12.13 | | |
| 시스템 개요 | **블루투스 비컨 기기와 라즈베리 파이를 이용한 반려동물 케어 장치**  - 하드웨어 : 라즈베리파이 B+, python3 language, 블루트스 비컨 (HM-10 모듈), 초음파 센서, 서보모터, 파이카메라  - 서버 : 소켓 통신 , mqtt broker  - 안드로이드 : 안드로이드 폰, Java language, 소켓 모듈  - 캣타워 : 우드락을 사용하여 모형을 제작  **제품 기능**  - 사용자가 집에 없을 때 혼자 남은 반려동물을 케어하는 장치  - 어플리케이션과 라즈베리파이가 통신하여 데이터를 주고 받는다.  - 어플리케이션에는 play, feed, find 세 버튼이 존재한다.  - play 버튼을 누르면 캣타워에 부착된 모터가 작동해 모터에 달린 고양이 장난감이 움직여 고양이와 놀아준다.  - feed 버튼을 누르면 캣타워에 부착된 사료통의 뚜껑을 열어 고양이에게 먹이를 준다.  - find버튼을 누르면 캣타워가 반려동물의 위치를 찾아 움직여 반려동물의 영상을 송출한다.  - 반려동물의 위치는 비콘을 통한 실내측위로 구현하였으며 삼각측량법, 칼만필터를 이용해 정확도를 높인다.  **기존 제품과의 차이점**   * 기존에 집에 고정되어 있는 반려동물의 놀이기구인 캣타워에 바퀴를 달아 움직이는 기구로 만듦 * 반려동물을 지켜볼 수 있는 것에서 넘어서서 놀아주거나 먹이를 주는 등의 다양한 기능이 가능하다 | | |
| 시스템 설계 | - 움직이는 캣타워는 안드로이드, 라즈베리파이, 비콘 기기가 mqtt와 소켓 통신을 사용해 통신한다.  - 비콘 기기는 HM-10모듈과 아두이노를 사용해 AT-command를 통해 비콘화 시킨 후, 전원을 연결하여 사용한다.   * 라즈베리파이 기기 2개를 비콘 스캐너화 시킨 후 , 캣타워와 고양이로 사용한다. * 캣타워와 고양이는 비콘의 rssi신호를 받아 집안에서 자신의 위치를 알 수 있다. * 캣타워는 mqtt를 통해 고양이의 위치를 지속적으로 전송받는다. * 고양이는 mqtt를 통해 자신의 위치를 캣타워에 지속적으로 전송한다. * 사용자는 안드로이드 어플리케이션을 통해, 캣타워 기기를 제어할 수 있다. * 사용자가 고양이의 모습이 궁금하면, 비컨 정보를 통해 고양이의 위치를 추적하여 찾아갈 수 있다. * 또한, 파이 카메라 모듈을 통해 영상을 송출 받아 고양이의 모습을 직접 확인할 수 있다. | | |
| 시퀸스 다이어그램 | - 고양이 위치를 찾을 때, 위치를 전송하는 시퀸스 다이어그램    - 영상 스트리밍을 위한 시퀸스 다이어그램  - 구현한 기능에 대해 시퀸스 다이어그램을 작성 | | |
| 하드웨어 설계 | **비콘을 이용한 위치계산**   * HM-10모듈과 아두이노를 사용해서 AT-command를 통해 모듈을 비콘화 시킴 * D = 10 ^ ((TXpower-RSSI) / (10\*n) ) 식을 사용하여 거리 값을 계산 * 비콘은 방향값을 갖지 않고, 스칼라 값만 갖기 때문에 최소 3개의 비콘이 필요 * 삼변 측량을 이용하여 위치 좌표를 계산 * 삼변측량(三邊測量)은 [삼각측량](https://ko.wikipedia.org/wiki/%EC%82%BC%EA%B0%81%EC%B8%A1%EB%9F%89)과 마찬가지로 [삼각형](https://ko.wikipedia.org/wiki/%EC%82%BC%EA%B0%81%ED%98%95) [기하학](https://ko.wikipedia.org/wiki/%EA%B8%B0%ED%95%98%ED%95%99)을 사용하여 물체의 상대 위치를 구하는 방법이다. 하나의 변의 길이와, 양 끝의 두 각을 이용하는 삼각측량과는 달리 삼변측량은 목표의 위치를 알기 위해서 두 개 이상의 기준점과, 물체와 각 기준점과의 거리를 이용한다. 삼변측량만으로 2차원 면에서의 상대위치를 정확하고 유일하게 결정하기 위해서는 최소한 3개의 기준점이 필요하다. (위키백과)  * 비콘의 rssi 값의 오차를 보정하기 위해 칼만 필터를 사용해서 보정 * 추가적으로 더 정확한 보정을 위해 20개의 값을 받아 평균값을 사용하여 보정   **파이 카메라를 통한 영상 송출**     * Flask를 이용해 영상을 스트리밍할 서버를 구축한다. * picamera 라이브러리 모듈을 통해 파이카메라를 제어한다. * rotation 메서드를 이용해 화면을 180도 로테이션 시켜준다. * resolution 메서드를 이용해 해상도를 설정한다. * 파이카메라의 프레임을 연속해서 웹페이지에 보내 스트리밍한다. * 안드로이드 어플리케이션에서 웹 뷰를 띄워 영상을 스트리밍한다. | | |

<별표 3-2>

|  |  |  |  |
| --- | --- | --- | --- |
| **설계모델 명세서 : 상세설계** | | **날 짜** | **2019.12.23.** |
| **작성자** | **김유진, 장준혁, 장우진** |
| 프로젝트명 | 비컨 기기를 활용한 반려동물 케어 IoT 장치 | | |
| 프로젝트 기간 | 2019.09.02. - 2019.12.13 | | |
| 안드로이드  어플리케이션  code 설계   * Activity 구조 | [FIND 버튼 클릭 시]  스크린샷이(가) 표시된 사진  자동 생성된 설명 스크린샷이(가) 표시된 사진  자동 생성된 설명 | | |
| 안드로이드  어플리케이션  code 설계   * MainActivity | - 스크린샷이(가) 표시된 사진  자동 생성된 설명  버튼 클릭시 :  라즈베리파이와 소켓 통신을 통해 어떤 버튼을 클릭했는지 데이터를 주고 받는다.  스크린샷이(가) 표시된 사진  자동 생성된 설명  소켓 통신 구현 class :  소켓 class를 이용해 라즈베리파이 ip와 3000번 포트를 통해 소켓을 구현한다.  라즈베리파이에 어떤 버튼이 클릭되었는지 버튼의 정보를 송신한다. | | |
| 안드로이드  어플리케이션  code 설계   * SettingActivity | 스크린샷이(가) 표시된 사진  자동 생성된 설명  find 버튼 클릭시 파이카메라 영상 스트리밍 웹 페이지를 띄워 영상을 송출한다 | | |
| 파이카메라 스트리밍 서버 구축 code | 스크린샷이(가) 표시된 사진  자동 생성된 설명  파이카메라의 프레임을 캡쳐하는 코드  카메라 해상도를 300, 600으로 설정하고 로테이션 메소드를 통해 화면의 방향을 정 방향으로 맞춘다. 파이카메라 화면을 캡쳐하고 프레임을 바이트 단위로 변환한다.    get\_frame 을 요청하면 위에서 변환한 프레임을 반환한다.  스크린샷이(가) 표시된 사진  자동 생성된 설명  파이카메라 영상을 스트리밍할 서버를 flask를 통해 구축한다.  templates디렉토리 하위에 있는 index.html 파일을 렌더링한 후 url동적 생성을 이용해 파이카메라의 프레임을 html파일 내 이미지 태그 src로 response한다. | | |
| code 설계   * 삼변측량을 이용한 좌표 계산 | - 삼변측량  - 비콘 3개와의 거리를 바탕으로 삼변측량을 활용하여 캣타워, 고양이의 좌표를 얻는다.    - solEqP, solEqM  - 삼변측량에 필요한, 이차방정식의 해를 구하는 함수  - 이차방정식(ax^2+bx+c=0)의 계수(a, b, c)와 이전의 해(beforeN)를 인수로 받는다.  - 계수들을 이용하여 판별식을 구하고, 판별식이 0보다 크면 이차방정식의 해를 반환하고 0보다 작으면 이전 해를 반환         * tri\_survey 함수   - 삼변측량을 시행하여 좌표를 반환하는 함수  - 캣타워, 고양이와 비콘 3개와의 거리(r1, r2, r3)와 비콘간의 거리(d1, d2, d3), 이전 좌표(beforex, beforey)를 인수로 받는다.  - r1, r2, r3 세 값이 모두 수치적으로 가능한 범위의 값인 경우, 비콘 3개를 중심으로 r1, r2, r3를 반지름으로 하는 원을 그리고, 세 개의 원이 만나는 곳의 좌표를 구하고 그 값을 반환한다.  - r1, r2, r3 중 하나의 값만 비정상적이라면 비정상적인 값 하나를 빼고 나머지 2개의 값을 반지름으로 하는 원 2개를 그려 두 원의 교점을 찾는다. 2개의 교점이 나오는데, 만약 비정상적인 값이 정상적인 범위보다 큰 경우면 좌표값이 더 큰 교점을 고르고, 비정상적인 값이 정상적인 범위보다 작은 경우면 좌표값이 더 작은 교점을 고르고 그 값을 반환한다.  - r1, r2, r3 중 비정상적인 값이 2개 이상이면 이전 좌표를 반환한다. | | |
| code 설계   * 칼만필터 구현 | * 칼만필터   - 과거의 정보와 새로운 측정값을 사용하여 측정값에 포함된 잡음을 제거시켜 값을 추정하는 칼만 필터를 파이썬 코드로 구현     * 생성자   - 프로세스 잡음과 측정 잡음을 인수로 받고 칼만필터 클래스를 생성     * filter 메소드   - 칼만 필터 알고리즘을 활용하여 필터링 한 좌표값을 반환하는 메소드 | | |
| code 설계  - 좌표 계산 후 전송 | * 변수 설정   -좌표 계산을 위한 각종 변수 설정  -소켓 통신, 블루투스 통신, MQTT 통신을 위한 변수와 좌표값 계산, 칼만 필터를 위한 변수 등으로 구성     * 블루투스 통신으로 비콘과의 거리를 받고, macAddress를 통해 어떤 비콘과의 거리인지 구분      * 비콘으로부터 rssi값을 구하고 그 rssi 값으로 비콘과의 거리를 구한다. * 측정값의 평균값을 구하기 위해 총합과 카운트 값을 더한다.      * 카운트 값이 20이 되면 평균값 계산   - 고양이 라즈베리파이에서 X, Y좌표를 구하고 캣타워 라즈베리파이로 전송       * 캣타워 라즈베리파이에서도 캣타워의 X, Y좌표를 구하고 고양이 라즈베리파이에서 받은 고양이의 좌표와 캣타워의 좌표를 바탕으로 고양이와 캣타워가 집의 어느 방에 있는지 계산      * 캣타워와 고양이 사이의 거리를 계산하고 계산한 캣타워 위치와 고양이 위치, 캣타워와 고양이 사이의 거리를 MQTT 방식으로 전송 | | |
| 캣타워 위치 좌표   * 계산 결과 | * 비콘의 rssi 값의 오차를 보정하기 위해 칼만 필터를 사용해서 보정 * 추가적으로 더 정확한 보정을 위해 20개의 값을 받아 평균값을 사용하여 보정 * 보정 전 값과 보정 후 값      * 위치 좌표를 통한 거리 계산 | | |
| Multifin.py  code 설계 | **<서보 모터(바퀴), 초음파 센서 초기 셋팅 >**    **<소켓 통신을 위한 초기 설정>**      -소캣 통신 수신 부분  **<mqtt 통신 부분>**  1. 연결 설정    2. 메세지 수신 부분    3. 수신한 메세지 파싱     * 수신한 메세지는 json형식이다. * Load후 json 형식의 데이터(string)를 파싱한 후, 알맞은 자료형으로 변환을 하여 변수에 저장한다.   **<멀티 스레드를 사용한 메인>**     * 멀티 스레드를 사용하여 소켓 통신과 mqtt통신을 할 수 있다.   **<초음파 센서 거리 계산>**     * 송신 트리거 발생 -> 펄스 중단 * Start : 초음파 전송이 끝나는 전송 시간을 저장 * Stop : 초음파 수신이 완료될 때까지 수신 시간을 저장 * Duration : 수신시간에서 전송시간을 빼서 총 도달 시간을 산정 * Ultradis : 음파의 초당 이동속도(340m)를 이용하여 거리를 계산 | | |
| **Find 모드**  **주행 설계** | * 주행에 필요한 global 변수와 지역변수를 선언한다. * Out : 캣타워가 방에서 나오기 전에는 0이고, 나온 후에 1이 된다. * Turn : 캣타워가 주행을 완료하는 동안 총 2번 턴을 한다. 한번 할 때 마다 1씩 증가한다.      * 실시간으로 mqtt통신을 통해 받아온 값을 이용해 주행한다. * Micro : 초음파센서와 앞의 장애물의 거리가 5cm이하이면 후진한 후 계속 주행한다. * 방에서 아직 나가지 못한 상태(out=0)에서는 캣타워의 y좌표가 1이 될 때 까지 직진한 후, out의 값은 1로 바꿔준다      * 방에서 나온 상태(out=1)이고, 캣타워가 복도에 있는 상태이면 오른쪽으로 90도 회전한다.      * 회전 후, 직진하며 캣타워와 고양이 사이의 거리를 비교한다. * 거리가 멀어지면 180도 회전한 후, 다시 직진하며 거리를 확인한다. * 거리가 가까워지면, 알맞은 방향으로 가고있음을 의미하므로 계속 직진한다. * 캣타워와 고양이의 x좌표가 같아지는 순간이 오면 오른쪽으로 90도 회전한다.      * 회전한 후, 다시 직진하며 캣타워와 고양이사이의 거리를 비교한다. * 멀어지는 경우, 위의경우와 같이 180도 회전한 후, 직진한다. * 캣타워와 고양이의 좌표가 일치하는 순간 캣타워가 정지한 후, 주행이 완료된다. | | |