|  |
| --- |
| 2023年全國大專校院智慧創新暨跨域整合創作競賽企劃書 |
| 競賽主題：  □ 1. 物聯網與金融科技組  ■ 2. 智慧機器組  □ 3. 數位永續科技組  □ 4. 體感互動科技組 |
| 1. 創作主題    1. 題目   災難救助遙測機器人  D2RSR (Disaster Relief Remote Sensing Robot)   * 1. 實用功能描述   D2RSR基於開源系統[*ROS(Robot Operating System)*](https://www.ros.org/)開發，是專為災害應對設計的遙控感測機器人系統，由控制系統與至少一台機器人載具構成，具有以下功能：   1. 多元的搖桿控制：使用者可以透過搖桿精準控制D2RSR，實現任意弧線、原地旋轉等高難度的移動，使D2RSR能靈活穿梭在任何環境和地形，深入災害現場。另外，搖桿還能控制網頁介面的影像呈現，提供身臨其境的第一人稱操縱體驗。 2. 即時360度環景影像：D2RSR配備了六台圍繞車身的攝影機，除了提供六方位的即時影像，更進一步利用影像融合技術，創建即時360環景影像。這允許操控者獲得第一人稱視角，即時觀察現場狀況。 3. 即時地圖繪製：D2RSR能利用內建的光達繪製二維平面地圖，由於光達不依賴可見光，能在視野昏暗的惡劣環境精準繪製地圖，救難人員能依據地圖規劃救難路徑，為受困人員爭取寶貴的救援時間。 4. 即時載具保護：D2RSR內建的光達訊號不僅能用於繪製地圖，進一步分析後還能提供機器人即時避障保護，當光達偵測到機器人過於靠近障礙物，會阻止機器人繼續往該方向靠近，避免碰撞造成損壞。 5. 完整的資訊呈現：D2RSR具有專為任何人設計的網頁介面，能即時呈現包括影像和地圖等各項資訊，操控者不須具備任何專業知識或額外訓練，即可快速掌握。   D2RSR的功能組合旨在確保災害應對時，救援團隊能獲得最即時、最全面的現場資訊，提高救援的效率與精確性。   * 1. 作品與市場相關產品差異   市面上的「災難救助機器人」功能強大，但由於此類別不是消費級產品，其價格通常高昂且不透明。以某消防機器人450萬元為例，它是我們所知的少數有明確價格的案例，如此高昂的價格使得現階段的災難救助機器人產品難以大規模普及。然而，D2RSR選擇了一條不同的道路。D2RSR不僅是一台災害救助機器人，它更是一套開源系統的體現。我們的目標是提供一個系統，不受特定硬體的限制。只要硬體可以安裝ROS系統，它便能成為D2RSR的載具。除了其開源的特性，D2RSR更展現了巨大的價格優勢，以我們選擇的硬體為例，成本僅為45000元，與競品相距百倍。  基於此理念，我們選擇了凌華科技開發的ROSKY作為基本載具，這是因為ROSKY是通用的ROS開發平台，並選用了Xbox搖桿作為主要控制器，這是因為Xbox搖桿是市面上最普及的遊戲搖桿，最值得一提的是，使用者可以根據自己的需求和預算自由選擇載具和控制器。D2RSR的設計注重模組化和標準化通訊規範，以確保其在各種硬體上都有出色的移植性。相對於傳統的災害應對機器人，它們可能需要高昂的資金投入，D2RSR的這種靈活性不僅降低了使用門檻，還賦予使用者更多選擇自主性，讓他們可以打造出最符合實際需求的系統。  作為開源系統，D2RSR的所有[程式碼與相關檔案](https://github.com/Sanlin1030142/nextTeam.git)均在GitHub上公開，任何開發者都能自由使用或新增、修改功能，讓使用者可以依自己的需求進行定制，獲得完全客製化的解決方案。另外，由於D2RSR基於ROS系統開發，受益於全球ROS社群的支持，能夠持續地獲得技術更新和優化。  在真正的災難救助場景中，時間就是生命。D2RSR的開放性和模組化設計，確保了救援團隊可以利用最新、最先進的技術，最大程度提高救援效率和成功率，同時低廉的使用成本確保了D2RSR能夠被廣泛運用於救援行動，為每一位受困者爭取一線生機。 |
| 1. 創意構想    1. 理論基礎       1. 搖桿控制：Linux中內建了Xbox搖桿的驅動程式，透過開源專案[*joystick\_drivers*](https://github.com/ros-drivers/joystick_drivers.git)將驅動程式的原始訊號轉換為[*ROS Message*](http://wiki.ros.org/msg)，並實作按鍵映射實現對車輛和影像的控制。       2. 即時環景：針對六個鏡頭進行[棋盤格標定法](https://docs.opencv.org/3.4/d4/d94/tutorial_camera_calibration.html)進行影像畸變校正，並將校正後的影像利用加權演算法以及多頻帶融合演算法進行拼接，最終利用影像壓縮演算法(MJPEG)將影像串流至網頁介面播放。       3. 地圖繪製：基於ROS系統中的[*gmapping*](http://wiki.ros.org/gmapping)套件，其原理是透過SLAM（Simultaneous Localization And Mapping）技術，在未知環境中透過光達建立地圖，最終透過[*ROS Message*](http://wiki.ros.org/msg)將地圖傳送至網頁介面呈現。       4. 載具保護：分析光達訊號，基於滑動視窗法實作光達雜訊過濾演算法以及障礙物辨識演算法，在偵測到障礙物時令車輛停止，提供可靠的防撞功能。    2. 設計創新說明       1. 硬體獨立性：D2RSR不受特定硬體的限制，它支援任何已安裝ROS系統的設備。       2. 客製化能力：D2RSR允許使用者根據需求和預算自選載具和控制器，這種模組化和個人化組合為使用者提供了前所未有的彈性。       3. 即時技術創新：包括即時避障和即時環景的技術創新，將以往需要大量分析且運算密集的演算法優化並重構，大幅降低對硬體算力的需求並提高運算效率降低延遲，使即時成為可能。    3. 特殊功能描述       1. 即時避障：這項功能基於光達進行距離感測，自研障礙物辨識演算法，能在車輛過於接近障礙物時主動介入使車輛停止，同時為了降低光達雜訊對障礙物識別演算法的干擾，自研雜訊過濾演算法，能根據當前環境最佳化雜訊過濾強度，保證機器人在任何環境都能受到保護。       2. 即時環景：陣列化控制車載的六個魚眼鏡頭，利用先進的影像處理、影像融合和影像傳輸技術，校正魚眼鏡頭產生的邊緣扭曲，融合六個畫面的影像同時傳輸到操控介面，提供高達40fps、延遲低於100ms的環景影像。 |
| 1. 系統架構    1. 架構說明   D2RSR系統主要分為三個關鍵部分：ROSKY自走車端、伺服器端和使用者端。下面是每個部分的詳細描述和功能：   1. ROSKY自走車端  * 攝影機陣列：配備了一組由六個攝影機組成的陣列，以捕捉即時影像並傳遞給伺服器端。 * 地圖建構：使用[*gmapping*](http://wiki.ros.org/gmapping)和光達技術，能夠建構和更新自走車的環境地圖。  1. 伺服器端  * 棋盤格標定法：對自走車端傳送的影像陣列使用棋盤格標定法進行校正，以確保每個魚眼鏡頭帶來的畫面扭曲都被校正。 * 360度環景合成：以加權演算法及多頻帶融合等算法，合成六個攝影機的畫面成為一張完整的360度環景影像。 * 避障功能：通過分析光達訊號，計算並調整車輪控制訊號，實現自走車的避障功能。  1. 使用者介面端  * 搖桿控制：使用者可以使用搖桿控制網頁界面，也可以向自走車發佈車輪控制訊號。 * 即時監控：查看六個經過校正的攝影鏡頭的即時監控畫面。 * 360度環景呈現：觀察自走車周圍的完整360度實時環景。 * 地圖相關功能：具有地圖建構的開關控制和地圖儲存等選項。   透過這三大關鍵元件的協作，D2RSR系統實現了全方位的環境感知、智慧控制與使用者互動功能。這種設計最大化降低了對自走車硬體算力的需求，同時確保了在救援中，D2RSR能提供即時、準確的資訊，使它成為靈活且可靠的控制系統。    圖表 1 系統架構圖   * 1. 「人機介面設計」（UI）與「使用者體驗」（UX）設計   對於D2RSR這樣的災難救助機器人，人機互動顯得尤為重要。為了確保使用者能夠流暢地與機器人進行互動，我們特別注重人機介面（UI）和使用者體驗（UX）的精心設計。   1. 直觀網頁介面：D2RSR具備簡潔的網頁介面，呈現機器人的狀態與即時環境資訊。網頁中所有的功能與設計，皆遵循現代網頁設計的規範，大大減少使用者的認知負荷。另外，網頁能自適應裝置，無論使用者是通過平板、筆記型電腦或智慧型手機，D2RSR的介面都能提供一致的操作體驗。 2. 實時監控：使用者可以從六個不同的攝影機同時監看，或選擇單獨監看特定的一個，單獨監看時更可以透過搖桿輕鬆切換所需的鏡頭。 3. 即時環景：六個攝影機的畫面能夠拼接成一個完整的360度環景，使用者可以透過搖桿控制視角，全面掌握環境狀態。 4. 即時地圖功能：D2RSR的介面允許使用者開始或停止繪製地圖，並能夠隨時下載當前的地圖資料。同時，畫面右方將顯示機器人周圍的障礙物狀況，讓操作者能夠隨時調整機器人的移動策略。 5. 實時回饋：當機器人接近障礙物時，介面將立即發出警告，讓操縱者能夠及時做出反應。   這些功能設計都是為了確保在危機情況下，救援人員可以有效、安全地操作D2RSR，最大限度地保護人的生命和財產。    圖表 2 首頁    圖表 3 實時監控-多鏡頭    圖表 4 實時監控-單鏡頭    圖表 5 即時環景    圖表 6 地圖與障礙物偵測 |
| 1. 計劃管理  |  |  |  | | --- | --- | --- | | 工作階段 | 工作日數 | 工作內容 | | 1 | 6 | 專案內容規劃及可行性驗證 | | 2 | 20 | 建置硬體和軟體環境 | | 3 | 20 | 實作各項功能之原型 | | 4 | 14 | 優化性能及強化功能性 | | 5 | 14 | 整合性測試 | | 6 | 7 | 設計展示流程 |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 周次 | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | | 起始日期 | | 7/11 | 7/17 | 7/24 | 7/31 | 8/7 |  |  |  | | 工 作 階 段 | 1 | V |  |  |  |  |  |  |  | | 2 | V | V | V |  |  |  |  |  | | 3 | V | V | V |  |  |  |  |  | | 4 |  |  | V | V |  |  |  |  | | 5 |  |  |  | V | V |  |  |  | | 6 |  |  |  |  | V |  |  |  | |

|  |
| --- |
| 1. 修改舊作參賽說明   ■ 本專案開發之作品未使用團隊成員曾獲競賽獎勵之作品。 |
| 1. 軟體清單   1. 作業系統環境  ■ Linux  2. 主要開發程式語言  ■ C++ ■ Python ■ 其他 HTML5, JS, CSS  3. 專案支援語言(可複選)  ■ 中文  4. 開發環境  軟體：  (1) HTML5  (2) Java Script  (3) CSS  (4) C++ (g++ 7.5.0)  (5) Python 2.7.17  (6) ROS  硬體：  ubuntu 18.04  5. 專案成果預定授權條款  本專案開發產品授權條款使用MIT授權宣告。 |
| 1. 權力分配   █ 依著作權法第 40 條之規定，由參賽學生與指導教授均等共有。 |