**REQUIREMENTS DOCUMENT**

**RESCUE ROBOTS**



**gemaakt door:**

**Kay van der Zander**

**Daniel van de Besselaar**

**Armand Mahieu**

**Inhoud:**

- Voorwoord

- Wat moet de robot kunnen

- Hardware

- Koppeling

- Software

-MoSCoW Methode.

-bijlage

Voorwoord:

Dit Requirements document is voor het door ons ontwikkelde C programma en de C# applicatie

voor de RP6. Rp6 is een robot die door een doolhof moet kunnen rijden. een doel moet kunnen vinden en dan de snelste weg terug moet kunnen rijden.

* 1. Wat moet de hardware kunnen
* De hardware moet de robot kunnen laten rijden.
* Objecten en muren moeten waargenomen kunnen worden, en de informatie moet doorgestuurd worden naar de chip die de informatie omzet in actie. Bijvoorbeeld de robot rijdt recht op een muur af, voordat deze tegen de muur aan rijdt moet hij stoppen en draaien om een nieuwe richting te vinden waarin wel gereden kan worden.
* Daarnaast moet de robot de coördinaten opslaan waarlangs is gereden en uit die coördinaten de snelste weg uit het doolhof kunnen vinden. Dit alles moet uit gelezen kunnen worden nadat die alles is gebeurd.
* De POC’s laten verder zien hoe de hardware werkt.
  1. Wat moet de software kunnen
* Verbinding maken met de basis station.
* Hij moet zijn route bestand kunnen lezen.
* De tekening van hoe de RP6 gereden heeft wordt dan zichtbaar, met daarbij in een lijstje in volgorde wat hij deed.
  1. Wat moet de koppeling kunnen
* De RP6 moet kunnen communiceren met het basisstation. Hierbij gaat het om informatie doorspelen en voordat de robot een gebouw in gaat een briefing leveren.
* Ook moet achteraf een debriefing gegeven worden waardoor de robot alle informatie die het nodig heeft opgeslagen heeft en weet dat het alles door heeft gegeven wat nodig was.
* Zie bijlage 1 voor verdere diepgang.
  1. Moscow methode.

De moscow methode houd in dat je dingen gaat rangschikken wat belangrijk is en wat leuk is voor erbij. voor als er meer tijd is.

Software

* M - MUST: deze eis *moet* in het eindresultaat terugkomen, zonder deze eis is het product niet bruikbaar;

1. De rp6 robot moet verbonden worden met de computer.
2. De gebruiker moet een bevestiging krijgen.
3. De route bestand moet kunnen uit gelezen worden.
4. De route moet ook zichtbaar zijn voor de gebruiker. hoe dat hij gereden heeft. en ook de terug weg.

* **S** - SHOULD: deze eis is zeer gewenst, maar zonder is het product wel bruikbaar.

1. De route die de rp6 robot afgelegd heeft in een mooie grafische omgeving te zien is, waar ook te zien is waar hij tegen aan botste. dus je krijgt een heel beeld over waar er muren zijn. Dat als hij dat detecteert dat het programma. Dan ook die muren daar neer zet.

* **C** - COULD: deze eis mag alleen aan bod komen als er tijd genoeg is;

1. Het programma moet de routes die hij binnen haalt kunnen opslaan
2. De opgeslagen routes moet je dan ook terug kunnen zien.

* **W** - WON'T: deze eis zal in dit project niet aan bod komen maar kan in de toekomst, bij een vervolgproject, interessant zijn

1. wat we er niet in gaan zetten is dat je een route real live kan zien.
2. Een weg kan kiezen als je de beelden toch real live ziet, dus dat je hem ook echt zelf bestuurt. dit zijn overigens wel aspecten wat je zou denken bij een rescure robot. dit is voor deze korte tijd niet echt reëel voor ons.

Software – Hardware koppeling

* **M** - MUST: deze eis moet in het eindresultaat terugkomen, zonder deze eis is het product niet bruikbaar;

1. Er moet een verbinding komen tussen het basisstation en de RP6 zodat er informatie kan worden uitgewisseld.
2. Er moeten sensoren bij komen die meten wat er gebeurd en hoe de robot rijd.
3. Bij de koppeling moet de robot de opgeslagen data die het heeft vervaardigd terug zenden naar het commando station, in dit geval een laptop.

De dingen die het moet doorsturen zijn:

* 1. de route die is gevolgd
  2. welke wanden voor, links, rechts gezien zijn
  3. waar het slachtoffer gevonden is

1. Doormiddel van de sensoren komt de robot aan de informatie die nodig is om aan het commando station door te geven waar het slachtoffer ligt.

De sensoren die de robot kan gebruiken zijn:

* 1. Druk sensoren aan de voorkant om te constateren dat er een obstakel voor de robot staat.
  2. Infra rood sensoren om rondom de robot om de muren of obstakels om de robot heen te meten.
  3. Licht sensoren om te meten naar brand of in de brand staand materiaal.
* **S** - SHOULD: deze eis is zeer gewenst, maar zonder is het product wel bruikbaar;

1. Er zal ook nog een koppeling zijn tussen de robot en een infra rood zender.
2. De robot kan bij dit project namelijk worden bestuurd door een afstandsbediening.
3. Hierbij zal een sensor gebrukt worden die infra rood zal ontvangen.
4. In de robot zal dan het signaal worden omgezet in code, die aantuurt om bepaalde dingen te doen die vooraf zijn ingesteld wanneer in bepaalde knop word ingetoetst.

* **C** - COULD: deze eis mag alleen aan bod komen als er tijd genoeg is;

Er zou op het lcd schermpje te zien kunnen zijn hoeveel meter de RP6 heeft moeten rijden.

* **W** - WON'T: deze eis zal in dit project niet aan bod komen maar kan in de toekomst, bij een vervolgproject, interessant zijn

Hardware

**Hardware MoSCoW**:

Must Have:

* **RP6 Base:** Zonder de base kan de rest niet draaien daarom is dit een must have
* **RP6 Control:** Hier komen de meeste sensoren/ledjes e.d. op dus een must have
* **Bumper** **Sensor**: Als de bumper sensor ingedrukt wordt moet de RP6 stoppen en een andere actie maken.
* **IR Sensor**: Moet reageren op de afstandsbediening
* **IR communication**: Regelt dat de binnenkomende opdrachten van de IR sensor doorgegeven worden aan de daarvoor bestemde onderdelen.
* **Status LEDS**: De LEDs moeten de status waarin de robot zich verkeerd aangeven.
* **Atmega32 microcontroller**: Hier wordt het programma opgeslagen
* **Motor current sensor**: Deze stuurt de motor aan en zal zo geprogrammeerd moeten worden.
* **5v voltage regulator**: Deze regelt de spanning over alle onderdelen.
* **New Gearing System**: Deze wordt aangedreven door de motor en laat de RP6 rijden.
* **Start Stop button**: Hiermee moeten we de robot kunnen starten en stoppen.
* **Mosfet H-Bridges:** Hiermee worden de Base en de control gekoppeld
* **Sharp IR analog Distance sensor:** Deze meet de afstand tot een object en zal zorgen dat onze robot niet te dicht bij de muren komt.

Should Have:

* **Light** **Sensor**: Als het te donker wordt (waarschijnlijk doordat De RP6 op een muur af rijdt) moet er een andere richting gekozen worden.
* **L2c Electronic Compass**: omdat de robot met alle andere sensoren het doel haalt is deze niet zo zeer een must maar we willen hem wel draaiend krijgen.

BIJLAGE 1:

[Communicatie Protocol](https://fhict.fontys.nl/IT/P4/T21c-11124/Shared%20Documents/Documentatie/Communicatie%20protocol.doc)