**Logboek Floris**

**Dinsdag 10 Juni:**

Bespreken en bedenken van het project onderwerp/onderzoek

Mbed online compiler repository gemaakt en gedeeld met de rest van de groep zodat iedereen bij de C++ code kan.

Gekeken naar de 3-wiel robot en proberen wat code te produceren om hem te laten bewegen.

**Woensdag 11 Juni:**

Een uur later aanwezig ivm een aanrijding met een persoon van de trein.

Verder gegaan met het proberen de robot aan te sturen. Uiteindelijk gelukt. De C++ code is gekalibreerd voor de robot met een neutrale pulsewidth van 0.001508 sec (staat in Servo.cpp).

De wagen is nu aanstuurbaar met WASD (via TeraTerm).

**Donderdag 12 Juni:**

Robot snelheid instelbaar (met range), en 45° draaiing mogelijk (range=0.0003, wait(3))

Robot met app bestuurbaar gemaakt en verder afgesteld.

Onderzoeken hoe de mobiel verbinding kan maken met de mbed, zonder dat de mbed stroom ontvangt via de USB.

**Vrijdag 13 Juni:**

Edwin is er niet en heeft de sleutel van de kast, dus we kunnen niet bij de robot. Ik kan dus niet uittesten om de robot draadlozer te maken. Wat informatie op internet gezocht over de power supply van de robot/mbed, en vector drive.

<http://books.google.nl/books?id=a-aWgVmA-x0C&pg=PA20&lpg=PA20&dq=mbed+lpc1768++power+for+usb+connection&source=bl&ots=X1ALyITPU0&sig=HQi9043lItcyad3H9pjr9B5hNSM&hl=nl&sa=X&ei=Q7iaU5jQFsj6PIfXgGg&ved=0CCEQ6AEwADgK#v=onepage&q=mbed%20lpc1768%20%20power%20for%20usb%20connection&f=false>

<http://www.societyofrobots.com/robot_omni_wheel.shtml>

<file:///C:/Users/Floris/Documents/3deec522f23a113c0a.pdf>

**Maandag 16 Juni:**

De robot “wireless” gemaakt, er gaan nu geen kabels meer naar stopcontact of PC. De motors krijgen stroom via de 4 batterijen, de USB wordt gepowered via een portable USB oplader.

2e robot aangesloten, en proberen om het MAC adres van de mbed te gebruiken om de robots te onderscheiden, omdat elke robot anders gecalibreerd moet worden.

Thuis nog even de mac adres arrays om mee te vergelijken gefixt.

**Dinsdag 17 Juni:**

Het onderscheiden van de robots mbv het mac-address werkt. De tweede robot is gecalibreerd (de neutral value, en het 45° graden draaien (op batterij). Het loodrecht laten rijden van de robots is zo goed als onmogelijk vanwege de vele variabelen die er invloed op hebben (grip, snoeren, energie niveau van de batterij) dus hier besteden we niet te veel tijd aan. Voor de demo corrigeren we eventueel met de hand. Als er tijd over is kunnen we mogelijk nog een Line-follower implementeren.

**Woensdag 18 Juni:**

De Mbed verder gecalibreerd en gewerkt aan het rijden van de juiste afstanden, robot delen vastzetten/opbergen en ordenen. De stroom toevoer wil nog steeds niet goed werken, en moet dus via stopcontact en laptop geleverd worden.

Een schema/tekening gemaakt van de opstelling van de robot en mbed, en de verbindingen.

**Donderdag 19 Juni:**

De stroomtoevoer van de robot/mbed wil nog steeds niet goed werken. De stroomtoevoer voor de robot werkt over het algemeen (indien de batterijen goed zijn opgeladen), maar de stroomlevering voor de USB werkt niet via de portable USB oplader en zal dus via een laptop of stopcontact moeten gedaan worden.

De code van de mbed is voorzien van een paar comments, en opgeschoond. Bij het importeren van het mbed programma moet men er aan denken dat er een aangepaste versie van de Servo library wordt gebruikt. Deze aangepaste versie is ook te vinden in mijn mbed.org repository.

Bij het demonstreren van de robot en besturing, kwamen Toto en Edwin met het idee om de master niet op een centraal device te draaien, maar hem te implementeren op de slave devices. Zo is steeds een ander device de master. Als een slave zijn bestemming al heeft gevonden, en dus inactief is, kan deze de rol van master op zich nemen. Dit zorgt voor een efficiënter gebruik van resources, en voorkomt dat als de master crasht, het hele netwerk plat ligt.

Door middel van Intents en startActivity() is het mogelijk om vanuit de ene app, een andere app aan te roepen en te starten. Dit maakt het mogelijk om bijvoorbeeld vanuit de slave app, de master app op te starten als de slave idle is. Dit brengt alleen het probleem met zich mee dat er onderhandelt moet worden of de slave de master rol wel kan innemen. Als de huidige master bezig is, kan zijn rol niet zomaar worden afgenomen. Ook zal dan alle informatie van de vorige master moet worden overgezet naar de nieuwe master, en alle slaves, zullen hun verbinding moeten verbreken met de oude master, en verbinding maken met de nieuwe master (wie laat je wel en niet toe??).

In het geval dat een master crasht, moet dit geconstateerd worden. Moet het systeem door gaan met een nieuwe master? En zo ja, hoe zorg je ervoor dat de informatie van de master niet verloren gaat, en kan worden gebruikt door de nieuwe master. Als de informatie niet kan worden gered, dan is het zinloos om door te gaan, omdat de bestemmingen van de robots niet meer bekend zijn. In dit geval is het voorkomen dat het netwerk plat ligt als de master crasht dus geen reden om te kiezen voor een dynamische master. Aangezien het systeem niet door kan gaan, ook al is er een nieuwe master.

**Vrijdag 20 Juni:**

Een beetje verder ingelezen over het dynamisch maken van de master. Verder was het niet een hele productieve dag.

**Maandag 23 Juni:**

Samen met Mike en Patrick hebben we het mogelijk gemaakt om meerdere slaves te verbinden met de master/gui. En werkt het om ook te receiven van deze slaves, zonder dat zij elkaar in de weg zitten.

Ook heb ik in de gui de array voor bezette locaties aangepast zodat deze makkelijk te gebruiken is, en biedt daarbij ook de optie om dit array te gebruiken voor het tekenen van de robots.

**Dinsdag 24 Juni:**

Verder gegaan met het programmeren van de gui/master. En hier en daar de rest geholpen met het oplossen van bugs en het implementeren van features in de gui. De tweede robot heb ik ook rij klaar gemaakt dus deze kan nu ook gebruikt worden. De code van de gui is wat opgeschoond en de grootte van het veld is nu makkelijk aan te passen, alleen de global x en y hoeven te worden verandert, en de rest schaalt vanzelf mee.

**Woensdag 25 Juni:**

De gui werkt nu naar behoren met verschillende kleuren voor verschillende robots. Door middel van het meegeven van de socket waarvan data wordt ontvangen aan de parser, wordt nu de juiste value in de robots dictionary geüpdate. Het probleem was dat eerst de “addr” variabele werd meegegeven, maar deze is global en bevat steeds het adres van het laatst verbonden device. Hierdoor werd steeds dezelfde key gebruikt, en dus wordt de dictionary onjuist geüpdate. Nu wordt de kleur bepaalt bij het verbinden van een device, en de rest van de values in de dictionary wordt in de parser geset.

**Donderdag 26 Juni:**

Klein beetje de gui.py code opgeschoond. Ook kwam ik er achter dat de mbed van de tweede robot (mbed 004) niet naar behoren werkte, zonder stroom van de adapter verbond hij niet met de computer, dit kan ook een van de redenen zijn waarom hij eerder ook al niet wilde werken met de portable USB oplader.

De gui print nu de randen van een gevonden vakje dat een eindbestemming is met de kleur van de robot waarvoor dit de eindbestemming is.