|  |  |  |
| --- | --- | --- |
| Vervang dit door je werktitel van je project | | Versie: 1.0 |
|  | | Datum: 11-03-20 |
| KLAS | 1MCT5 | |
| Naam Voornaam | Van Den Hende Simon | |
| Naam Voornaam sparring partner  *(iemand van jouw klas)* | Everaert Alexander | |

|  |
| --- |
| Opdrachtstelling |
| Hack of bouw INDIVIDUEEL een DEVICE dat automatisch opstart waarbij data van de minimum 3 (waarvan 1 niet in de les behandelde) sensoren wordt verwerkt op een Raspberry Pi.  Voorzie ook minstens 1 actuator die je kan aansturen vanuit de website en ook volkomen autonoom kan werken.  De verwerkte data wordt opgeslagen in een MySQL/MariaDB op de Raspberry Pi. Gebruik de image uit full stack web development. We willen sowieso historiek weergegeven krijgen op de mobile-first website  Via een Flask backend op de Raspberry Pi wordt de data naar een HTML/ CSS/JS frontend op een aparte web server gestuurd. Voeg in Flask een route toe om de Raspi correct af te sluiten  Het IP-adres moet verplicht worden weergegeven op een display.  opm. een camera wordt afgeraden en kan enkel optioneel gebruikt worden nadat alles is afgewerkt. |

|  |
| --- |
| Randvoorwaarden |
| Maakbaar in 3 weken en 2 dagen, volledig gedocumenteerd en getest  Het project moet door een MCT student te hermaken zijn (= “re-creatable”)  Het project mag géén klakkeloze rip-off zijn van bestaande projecten.  Tip: Je hebt een **sparringpartner**. Zoals bij elke opdracht voor project1 is het de bedoeling dat hij / zij dit naleest en feedback geeft |

|  |
| --- |
| Projectresultaat**:** Wat is het als het klaar is?  Voor wie is je project bedoeld en wat doet het? |
| Mijn project is bedoeld om het banale en soms ouderwetse proces van kleren wassen te automatiseren.  Zo zal de gebruiker zich niet langer moeten bezighouden met   * Kleren sorteren * Aantal kleren in de wasmand bijhouden * Wasproduct afmeten / bijhouden   Daarenboven kan de gebruiker de wasmachine van op afstand bedienen. Zo kan hij ervoor zorgen dat de was gedaan is tegen dat hij thuis is.  De machine blijft natuurlijk ook offline bestuurbaar. De gebruiker kan zijn kleren handmatig in de trommel doen en vervolgens een programma kiezen op de ingebouwde LCD display aan de hand van drukknopjes. |

|  |
| --- |
| Functionele Eisen**:** Beschrijf je voorstel: licht volgende onderdelen toe.  Maak duidelijk hoe jouw voorstel beantwoordt aan de eisen van de opdracht |
| |  |  | | --- | --- | | Eis | Invulling | | 1. behuizing (maakgedeelte): 3D print? Lasercutting? Hacken van bestaand iets? | **Het wasmand-gedeelte** zal grotendeels bestaan uit dunne platen scrapwood, aangezien de meeste onderdelen van dit gedeelte bestuurd worden door servo motors met een beperkte draagkracht. Het gewicht van de kleren moet opgeteld worden bij de platen onderaan de gesplitste wasmand. Voor het prototype zal dit echter beperkt blijven. De statische delen zal ik in deftiger hout maken voor meer stevigheid en een mooiere look. Rond dit gedeelte zullen doorzichtige platen gemonteerd worden, deels voor de visualisatie van het onderliggende mechanisme, maar ook voor directere feedback. Zo kan de gebruiker ook gewoon gaan kijken hoeveel was er in de mand zit, zonder zijn smartphone te moeten raadplegen.  **De wasmachine** zelf had ik graag gehackt van speelgoed, maar die zijn allemaal front-load machines. Voor dit project heb ik expliciet een top-load machine nodig. Deze zal ik dus zelf maken, weliswaar zeer primitief. Ik ga een emmer nemen en een DC motor monteren aan de onderkant, deze zal voor de omwentelingen zorgen. De maximale snelheid voor dit prototype blijft beperkt, dus ik denk niet dat ik extra ondersteuning moet inbouwen. De behuizing daar rond zal bestaan uit houten plankjes. De bovenkant kan open en toe gezet worden met een servo motor, dus deze zal weer uit dun hout gesneden worden. Het display wordt aan de voorkant gemonteerd samen met de push buttons voor manueel gebruik.      Alle kabels worden onzichtbaar naar de raspi gebracht die zich onderaan de constructie bevindt.  Onder de emmer zal ik een staander voorzien voor de DC-motor. Deze wordt ge3Dprint. | | 1. electronica:  2 gekende sensoren minimum 1 *nieuwe(1)* sensor  ((1)sensor die niet is behandeld in de lessen prototyping); 1 actuator: (tip: 1 ledje is niet voldoende als actuator!) 1 display: (tip: neem die uit je doos) | Bovenaan de constructie staat een **RGB-Sensor**. Deze zal bepalen tot welke categorie de kledij behoort. Aangezien in dit prototype de range van de sensor niet overschreden wordt indien de bak leeg is, zal ik nog een **Passieve infrarood bewegingsensor** monteren. Deze zal doorgeven of er kledij in de bak gegooid werd. Deze zal dus de RGB-Sensor trigerren om een meting te doen. Deze meting zal meerdere malen plaatsvinden, en de gemiddelde waarden worden doorgegeven om ruis te verminderen. Nadat deze meting verwerkt is, zal de juiste sequentie van **servo motoren** **(3)** in gang gezet worden om de kledij in het juiste vak te laten vallen. Wanneer de servomotoren terug op hun initiele positie staan, wordt een meting gedaan van de **Ultrasonische afstandssensor** van de desbetreffende bak, ook deze zal meerdere malen plaatsvinden en geresampled worden.  Deze metingen gebeuren dus enkel wanneer er kledij bijgekomen is. Onderaan deze bakken bevinden zijn ook **nog 2 servo motors**. Deze openen het luik onderaan, waardoor de was naar beneden kan vallen, in de trommel. Indien nodig, zal de pi het luik boven de trommel eerst open zetten, adhv zijn desbetreffende **servo motor**. Terwijl de kleren vallen, wordt de trommel rondgedraaid door een **DC motor** zodat de kleren evenredig verspreidt worden. Deze DC motor zal gedurende elk programma de trommel controleren. Tijdens het programma wordt ook het wasproduct automatisch toegevoegd via een **peristaltische pomp**. Ahdv een **moisture sensor** zal de gepaste hoeveelheid vloeistof geleverd worden. Via deze sensor kan de gebruiker ook het pijl bijhouden van zijn wasproduct. | | 1. datacaptatie (backend) berekenen en opslaan van wat je moet bijhouden om je ding te doen werken | De sequentie van de metingen staat hierboven beschreven. Hieronder staat mijn voorstel voor de tabellen en hun velden.  **tblSensor**   * SensorID * Label * Timestamp * Value (Bij kleurensensor geen RGB-Value maar 0 voor bleek en 1 voor gekleurd, omzetten met case / if structuur in de query voor visualisatie)   **tblWas**   * WasID (PK) * ProgrammaID (Relatie) * MandID (Relatie) * Timestamp   **tblProgramma**   * ProgrammaID (PK) * AmountOfProduct   + Moisture sensor zijn value opvragen eerst, anders kun je programma niet selecteren. (select value from tblSensor where SensorID = x) * Duration * Temperature   **tblWasmand** (CurrentLevel berekenen via value van de SensorID, niet opslaan)   * MandID (PK) * SensorID (Relatie) ( 1 vd distance sensors ) * Width * Depth * MaxHeight   + Checken voor de topservo motion in gang te zetten → kleren niet meer toevoegen als de mand al vol zit * BottomServo (pin op de raspi) * TopServo (pin op de raspi)   + Op basis van deze weet ik welke de andere top servo is en welke direction de middelste pillar moet draaien | | 1. visualisatie (frontend) tip: we willen sowieso historiek weergegeven krijgen op de mobile-first website | De landing page wordt gesplitst in 2 delen, bleke en donkere was. Op een desktop worden deze naast elkaar weergegeven, maar op mobile display ik er maar 1.      Met behulp van css en javascript zal ik dus heel veel UI elementen verplaatsen / (on)zichtbaar maken. Op deze manier wil ik het gevoel van een app creeeren terwijl ik in de browser blijf.  Op de desktop ziet de pagina er nog vrij sober uit. Het zijn dan ook zeer vroege schetsen. De whitespace is op zich niet slecht vind ik. Ik kan eventueel iets met de achtergrond doen zoals je ziet bij de mobile versie om het toch wat interessanter te maken.    Over de ‘new programme’ button in de nav bar ben ik ook nog niet helemaal zeker, maar ik weet niet goed hoe ik deze feature op een andere manier naar voor kan brengen. Ik ga dit nog eens langs mijn sparring partner runnen. | | 1. genormaliseerde SQL database (raspi = verplicht) | OK/NOK | | 1. webserver opzetten voor besturing van project - draait verplicht op de raspi | OK/NOK | |

|  |
| --- |
| Blokschema Aan de hand van dit blokschema is het duidelijk wat je gaat maken  Opgelet:  Raspi in het midden  Liefst IN links en OUT rechts tekenen tov de raspi  Sowieso: IN met pijl & OUT met pijl  *Zie voorbeeld uit de theorieles* |
|  |