



Smart Systems

PRESENTATIE POEMBAK DELCO



Meet the team

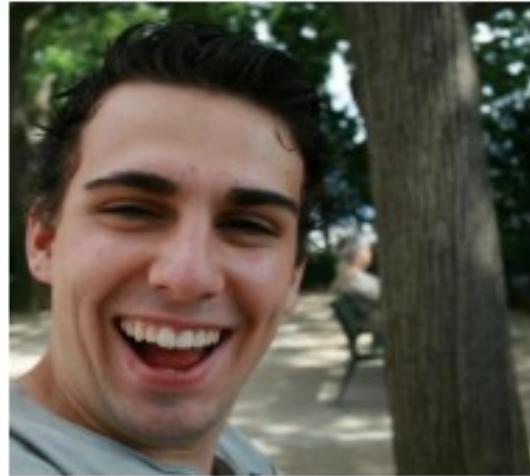
6 PEOPLE, 1 ROBOT

Team Besturing



Ski De Keersmaecker

Team Pomp



Koen Schauwaert

Team Emmer



Robbe Goethals



Ruven Salamon



Maarten Wachters



Wouter Peetermans



Progress

Waar staan we nu

- Concept uitgewerkt
- Eindhoven
- Concept verbeterd
- Toestemming
- Sponsoring CCM
- Materialen besteld
- Materialen geleverd

Waar staan we nu

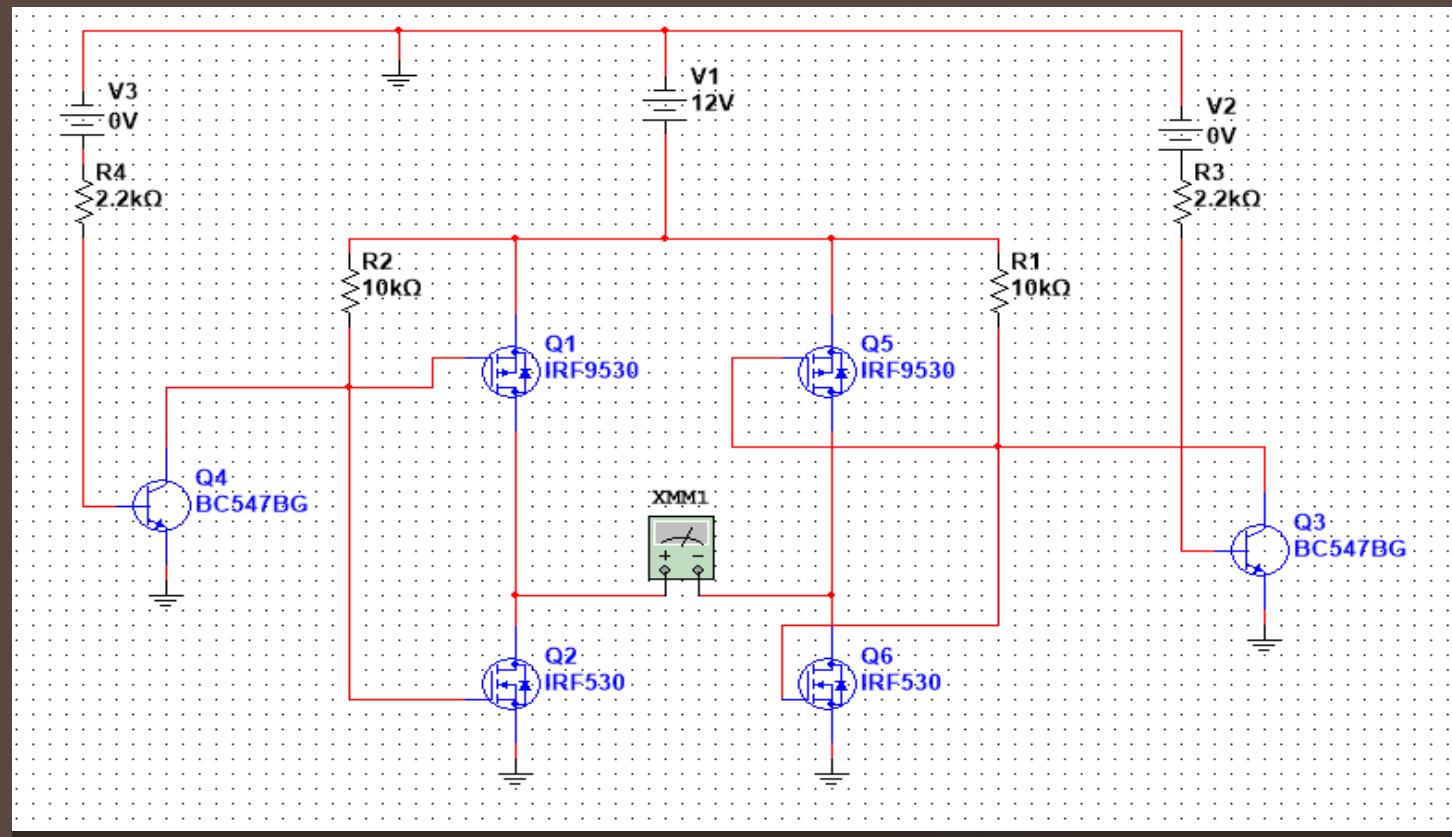
- Algemeen
- Besturing
- Pomp
- Emmer

A black and white photograph of a rural landscape. In the foreground, there's a road or path leading towards a small, isolated building, possibly a farm or a station. The background features rolling hills and fields under a clear sky.

Algemeen

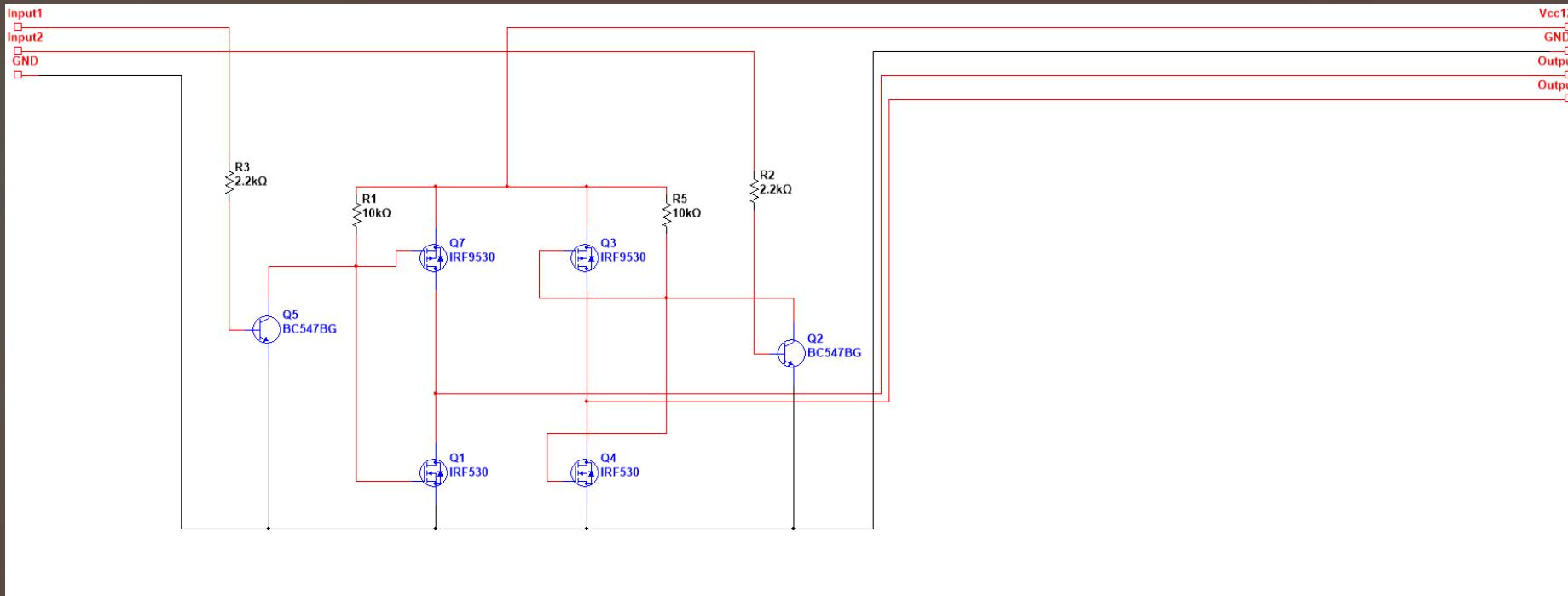
Algemeen

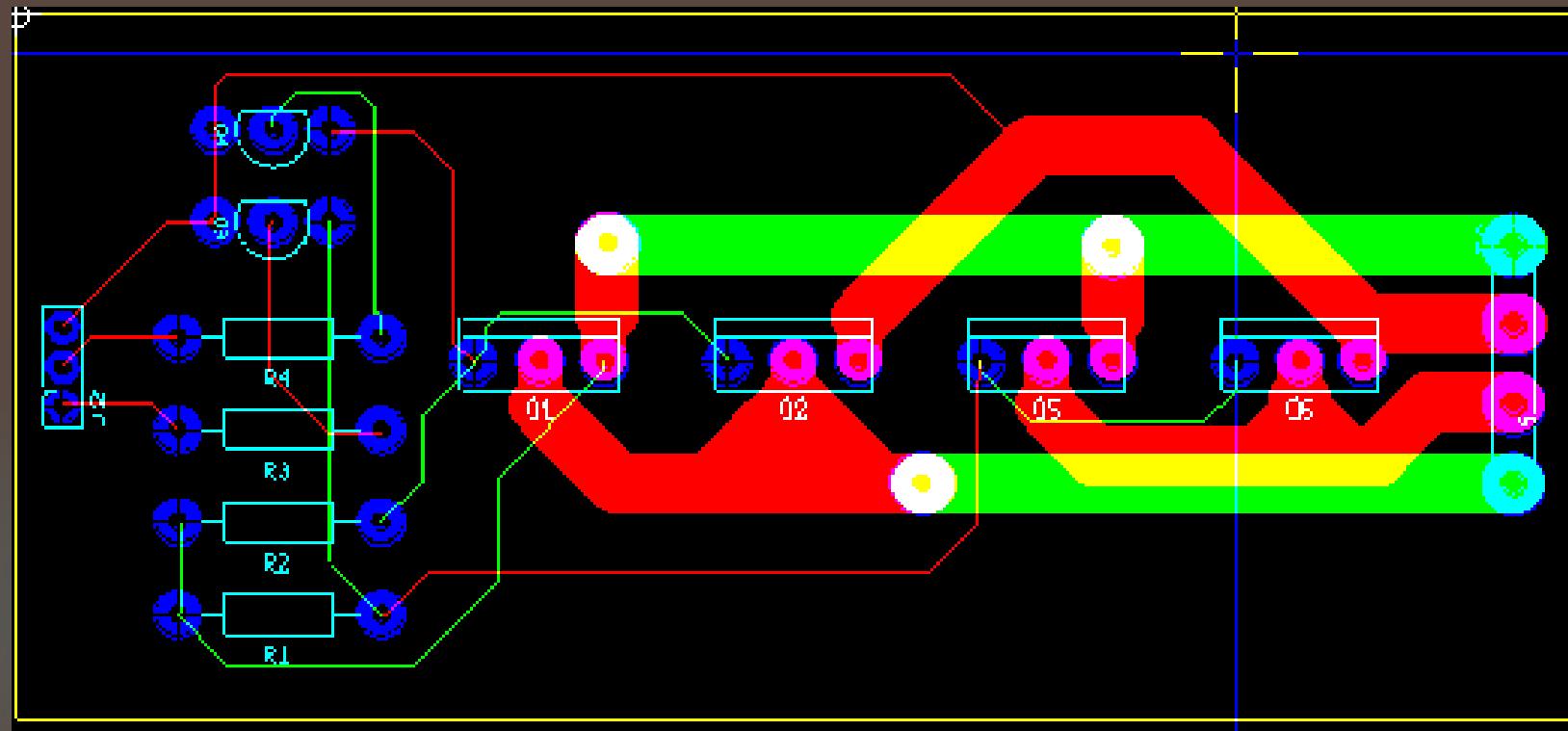
- ✓ Maken motorcontroller



Algemeen

- ✓ Maken motorcontroller
- ✓ PCB motorcontroller

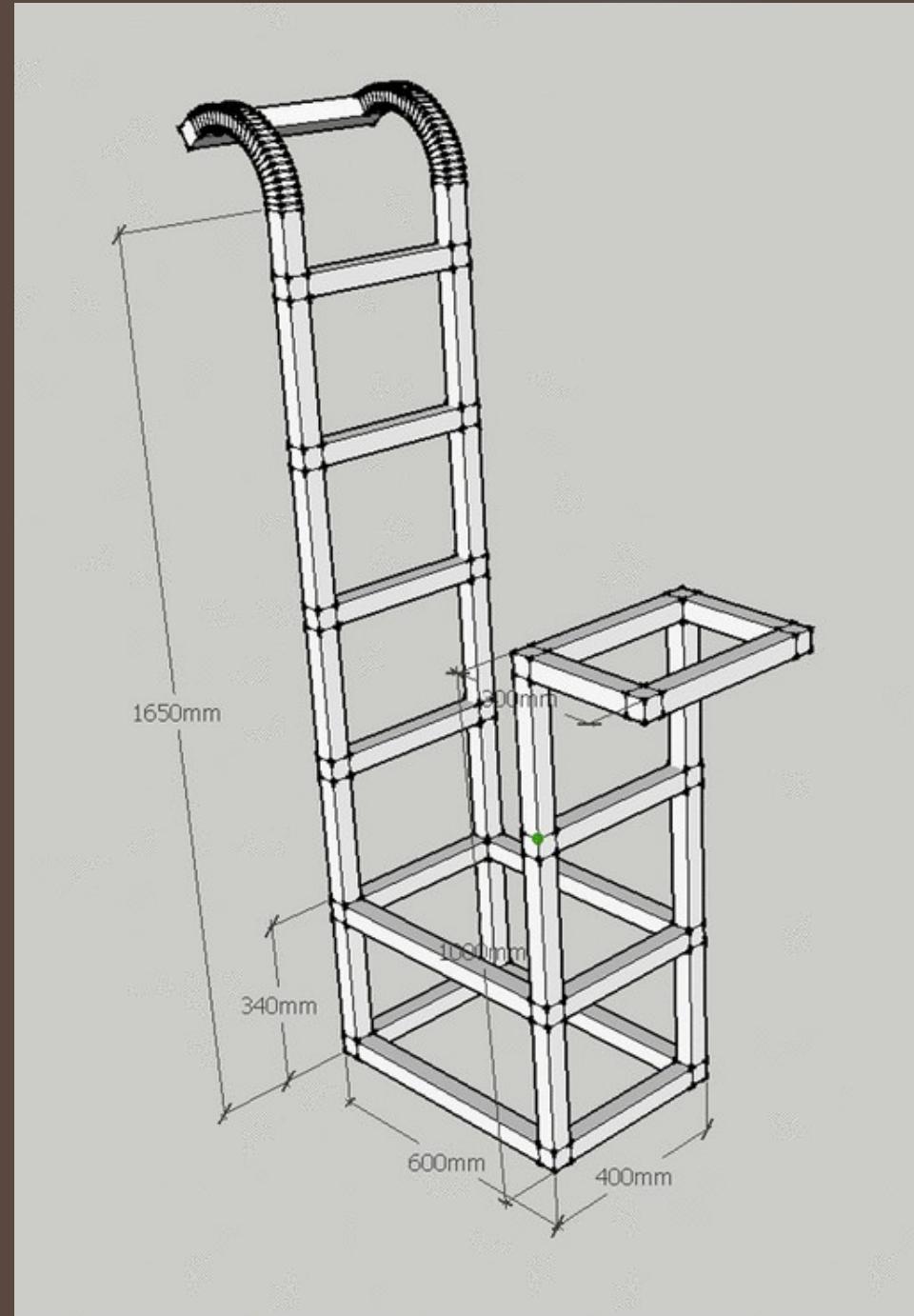


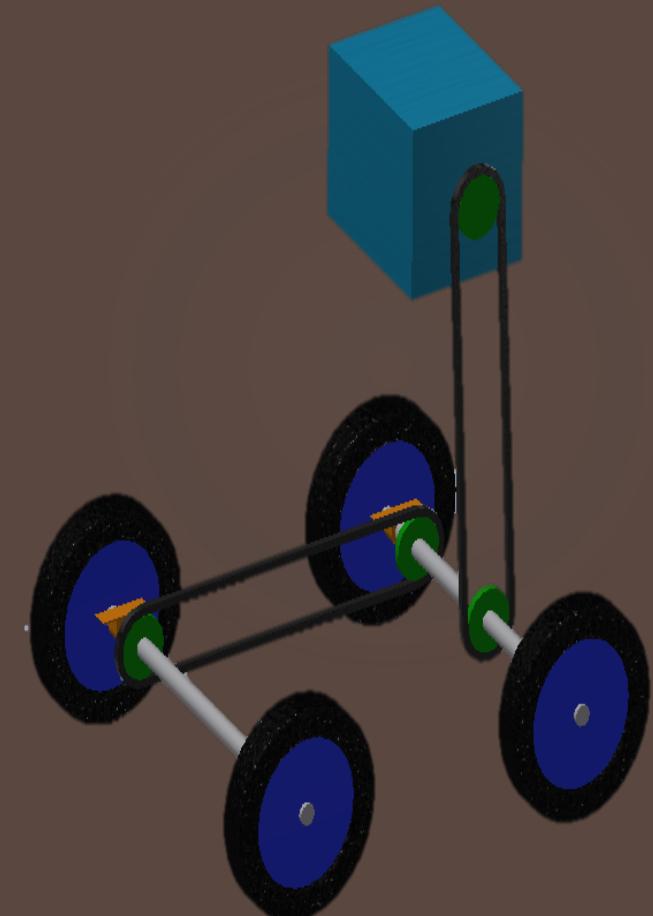
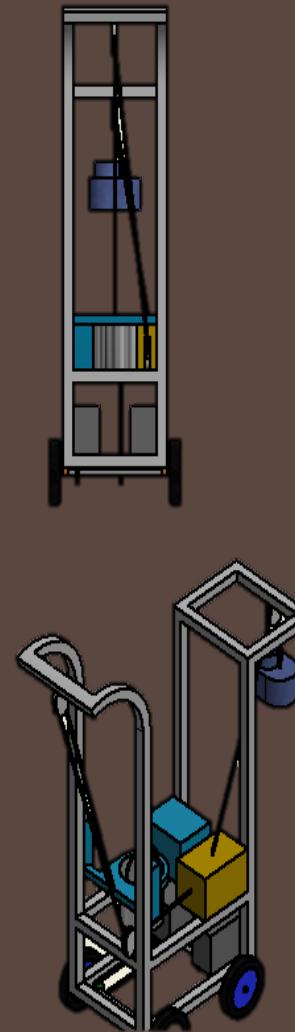
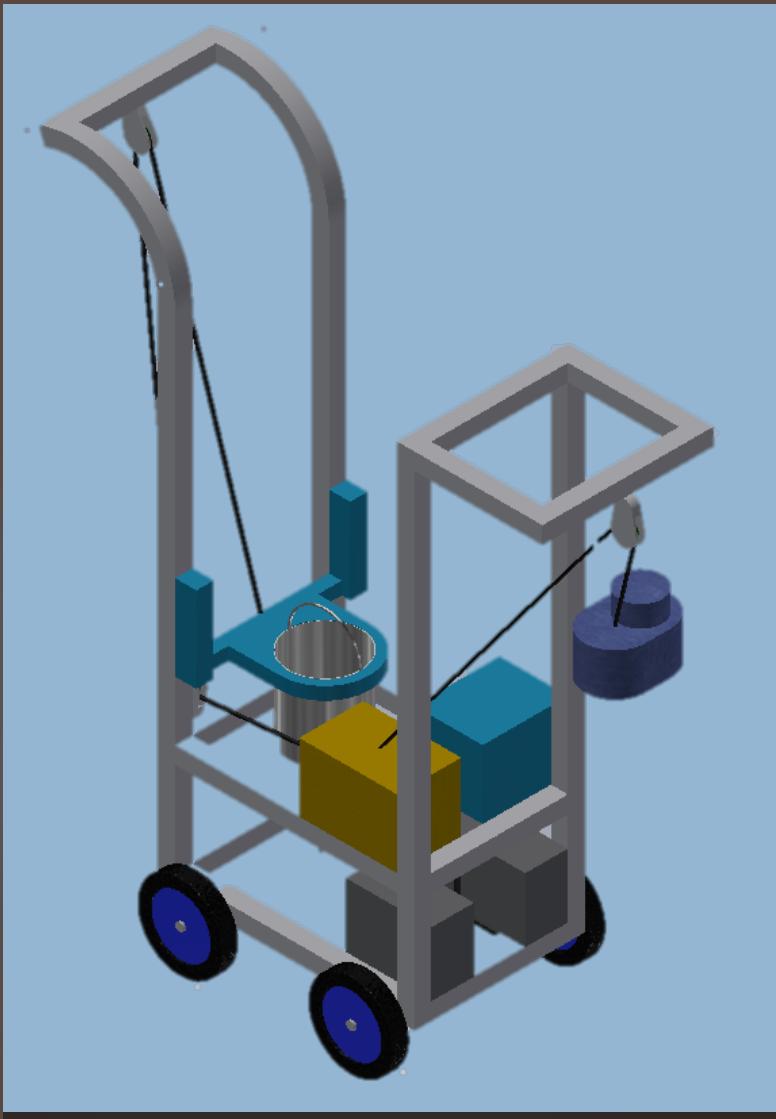


PCB motorcontroller

Algemeen

- ✓ Maken motorcontroller
- ✓ PCB motorcontroller
- ✓ Frame





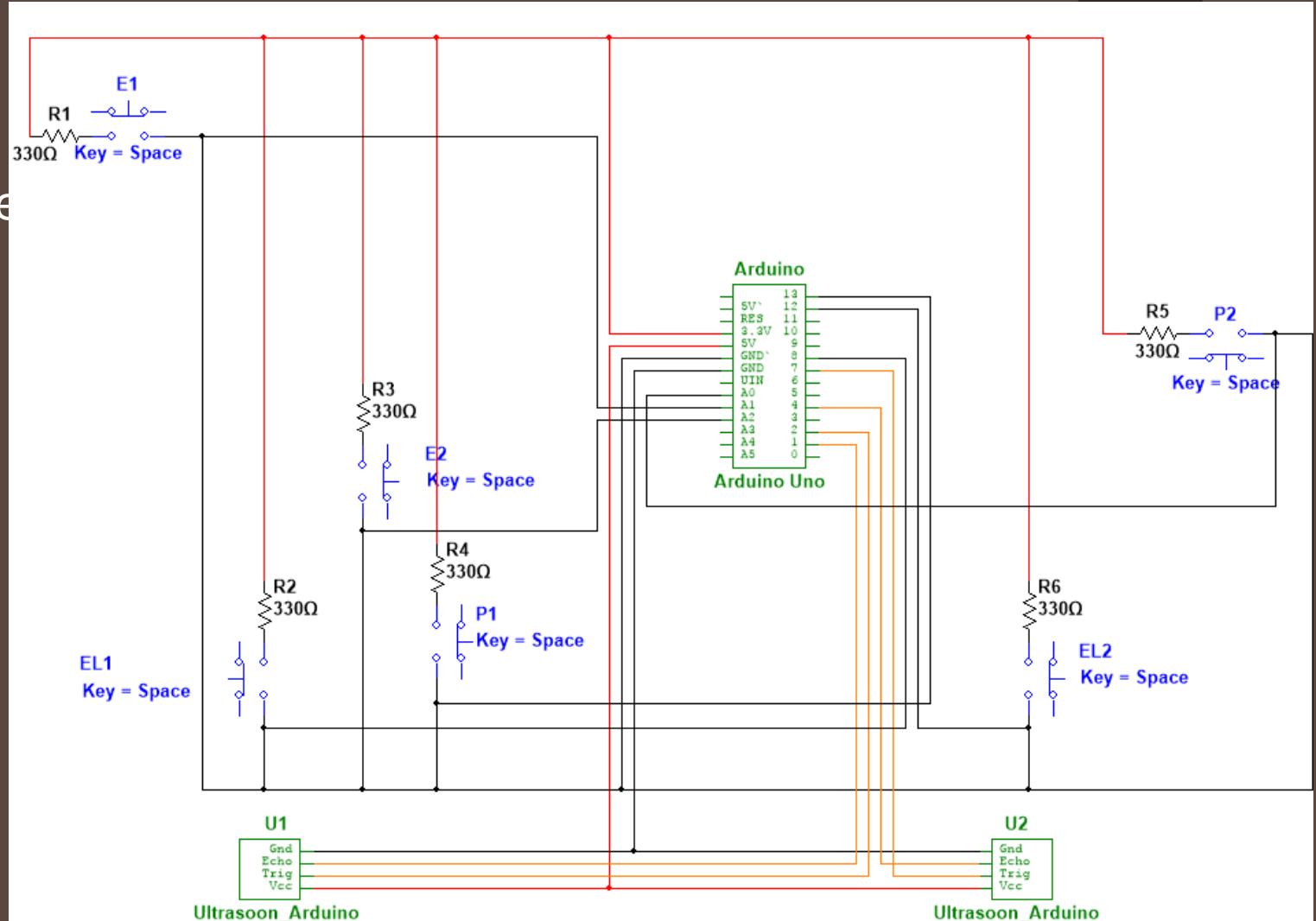
Algemeen

- ✓ Maken motorcontroller
- ✓ PCB motorcontroller
- ✓ Frame
- ✓ Testopstelling



Algemeen

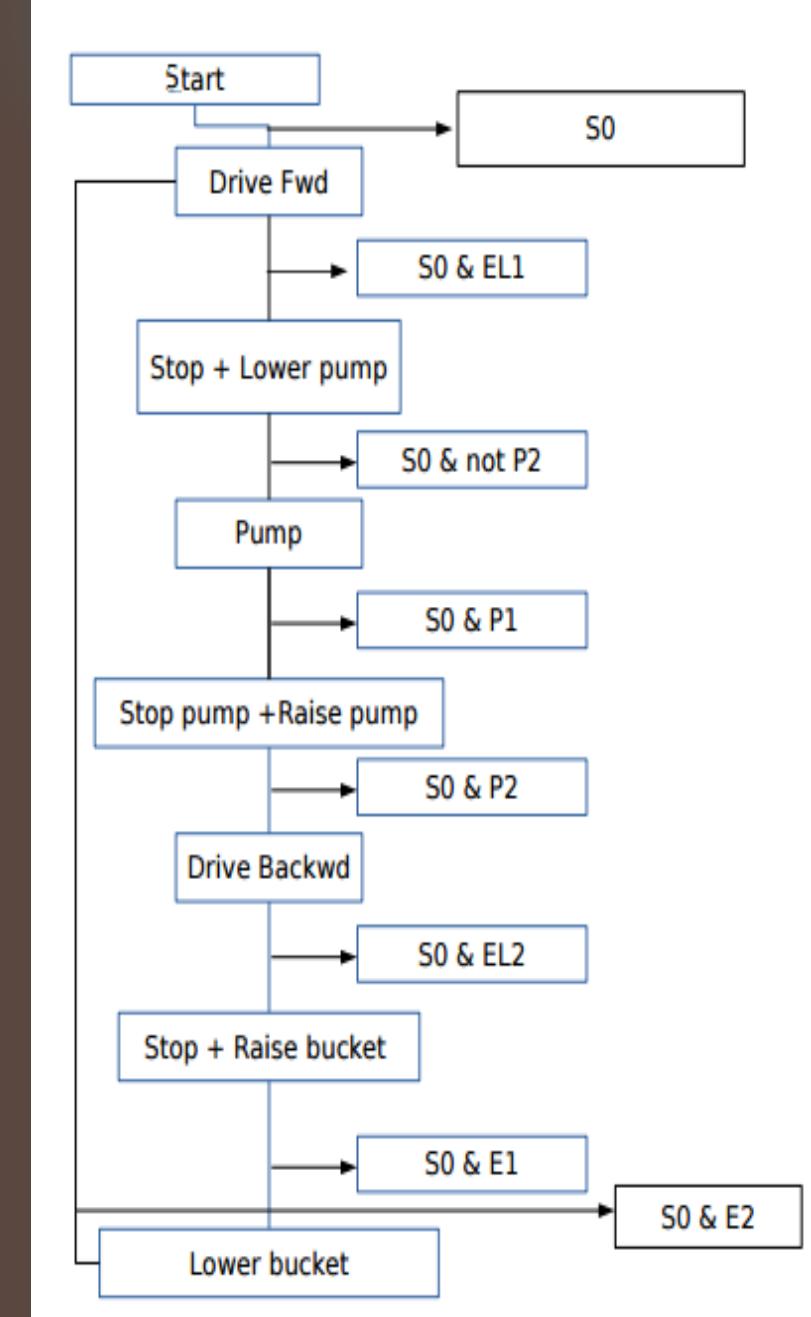
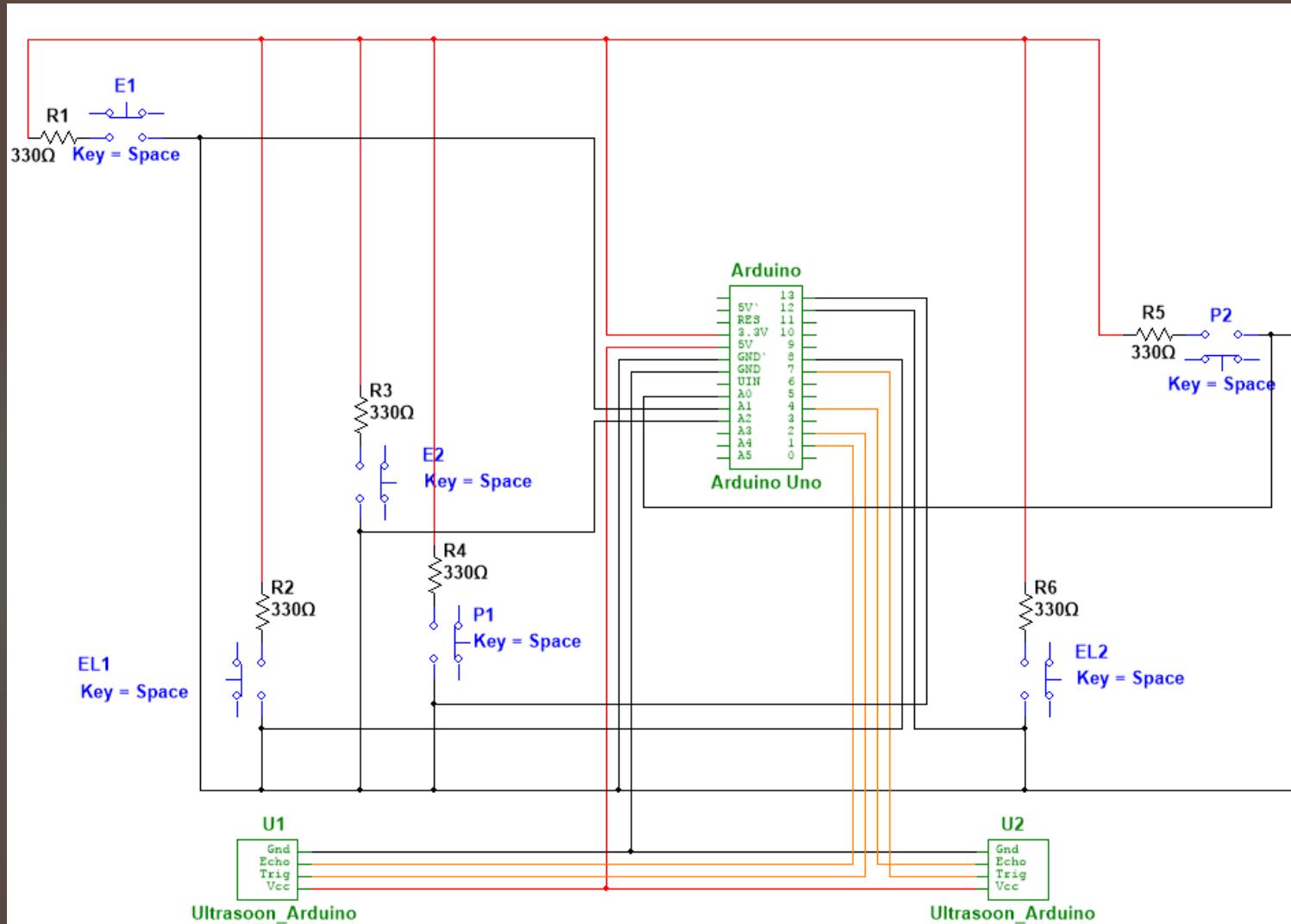
- ✓ Maken motorcontroller
- ✓ PCB motorcontroller
- ✓ Frame
- ✓ Testopstelling
- ✓ Sensoren inlezen



Algemeen

- ✓ Maken motorcontroller
- ✓ PCB motorcontroller
- ✓ Frame
- ✓ Testopstelling
- ✓ Sensoren inlezen
- ✓ Verloop proces - Graphset

Graphset verloop



Algemeen

- ✓ Maken motorcontroller
- ✓ PCB motorcontroller
- ✓ Frame
- ✓ Testopstelling
- ✓ Sensoren inlezen
- ✓ Verloop proces – Graphset
- ✓ Werking code volgens PLC



Code

Code

```
void setup() {  
    initSensorPins();  
    startComm();  
}  
  
void loop() {  
    inputScan(); //update status of sensors.  
    executeProgram(); //update state and perform action  
    callHome(); //send debugging information.  
    listenHome(); //see if we got instructions.  
}
```

```
void executeProgram() {  
    updateState();  
    stateAction();  
}
```

Code

► Toestand onthouden en sensoren inlezen

```
void updateState() {
    switch (currentState) {
        case Off:
            if(running){
                currentState = DrivingForward;
            }
        //Als 'aan' en eindeloopschakelaar pompzijde ingedrukt
        case DrivingForward:
            if (sensorMask & ELOOPPOMP && running) {
                currentState = LoweringPump;
            }
            break;
        //Als 'aan' en pompsensor emmer zijn ingedrukt (dus emmer is vol)
        case LoweringPump:
            if (sensorMask & POMP1 && running) {
                currentState = RaisingPump;
            }
            break;
    }
}
```

```
//Als 'aan' en pompsensor bassin zijn ingedrukt (dus pomp is opgehaald)
case RaisingPump:
    if (sensorMask & POMP2 && running){
        currentState = DrivingBackwards;
    }
    break;

//Als 'aan' en eindeloopschakelaar emmer zijde ingedrukt (robot tegen vuur zijde)
case DrivingBackwards:
    if (sensorMask & ELOOPENMER && running) {
        currentState = RaisingBucket;
    }
    break;

//Als 'aan' en emmer sensor bovenaan ingedrukt (emmer is dus opgehoffen)
case RaisingBucket:
    if (sensorMask & EMMER1 && running) {
        currentState = LoweringBucket;
    }
    break;

//Als 'aan' en emmer sensor onderaan ingedrukt (emmer terug gedaald)
case LoweringBucket:
    if (sensorMask & EMMER2 && running) {
```

Code

- ▶ Toestand onthouden en sensoren inlezen
- ▶ Verandering van staat

```
void stateAction() {  
    switch (currentState) {  
        case DrivingForward:  
            drive(true);  
            break;  
  
        case LoweringPump:  
            lowerPump();  
            break;  
  
        case RaisingPump:  
            raisePump();  
            break;  
  
        case DrivingBackwards:  
            drive(false);  
            break;  
  
        case RaisingBucket:  
            raiseBucket();  
            break;  
  
        case LoweringBucket:  
            lowerBucket();  
            drive(true);  
            break;  
    }  
}
```

Code

- ▶ Toestand onthouden en sensoren inlezen
- ▶ Verandering van staat
- ▶ Sensoren aan Arduino

```
int getSensorMask() {
    int mask = 1;

    for(int i = 0; i < NUM_PINS; ++i){
        if(digitalRead(sensorPins[i])){
            mask |= 0x01 << i;
        }
    }

    return mask;
}
```

```
#define ELOOPEMMER 0x01
#define ELOOPPOMP 0x02
#define POMP1 0x04
#define POMP2 0x08
#define EMMER1 0x10
#define EMMER2 0x20

void initSensorPins();
int getSensorMask(); //returns a mask indicating which sensors are
void measureDistances();
```

Code

- ▶ Toestand onthouden en sensoren inlezen
- ▶ Verandering van staat
- ▶ Sensoren aan Arduino
- ▶ Debugging

```
void callHome() {  
    send(&sensorMask, sizeof sensorMask);  
    send(&currentState, sizeof currentState);  
}
```

Code

- ▶ Toestand onthouden en sensoren inlezen
- ▶ Verandering van staat
- ▶ Sensoren aan Arduino
- ▶ Debugging
- ▶ Instructies

Code

- ▶ Toestand onthouden en sensoren inlezen
- ▶ Verandering van staat
- ▶ Sensoren aan Arduino
- ▶ Debugging
- ▶ Instructies
- ▶ Simulatie werkt



A black and white photograph of a rural landscape. In the foreground, there's a road or path leading towards a small, isolated building, possibly a farm or a station. The background features rolling hills and fields under a clear sky. The image has a slightly grainy texture and a color palette dominated by blues and greys.

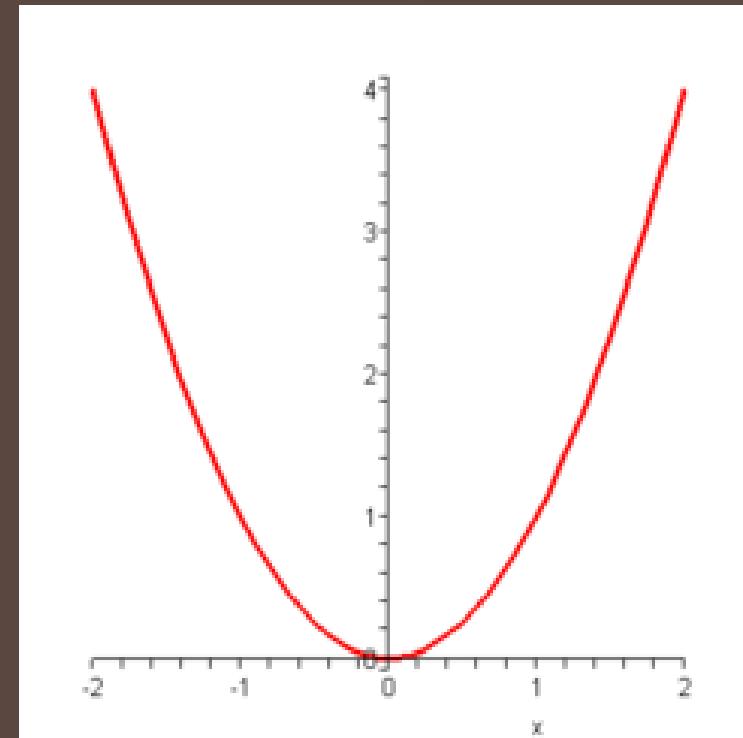
Besturing

Besturing

- ▶ Nog geen wielen
- ▶ Ultrason sensoren voor en achter

Besturing

- ▶ Nog geen wielen
- ▶ Ultrason sensoren voor en achter
- ▶ Algoritme
 - ▶ Verloop functie rijden





Pomp

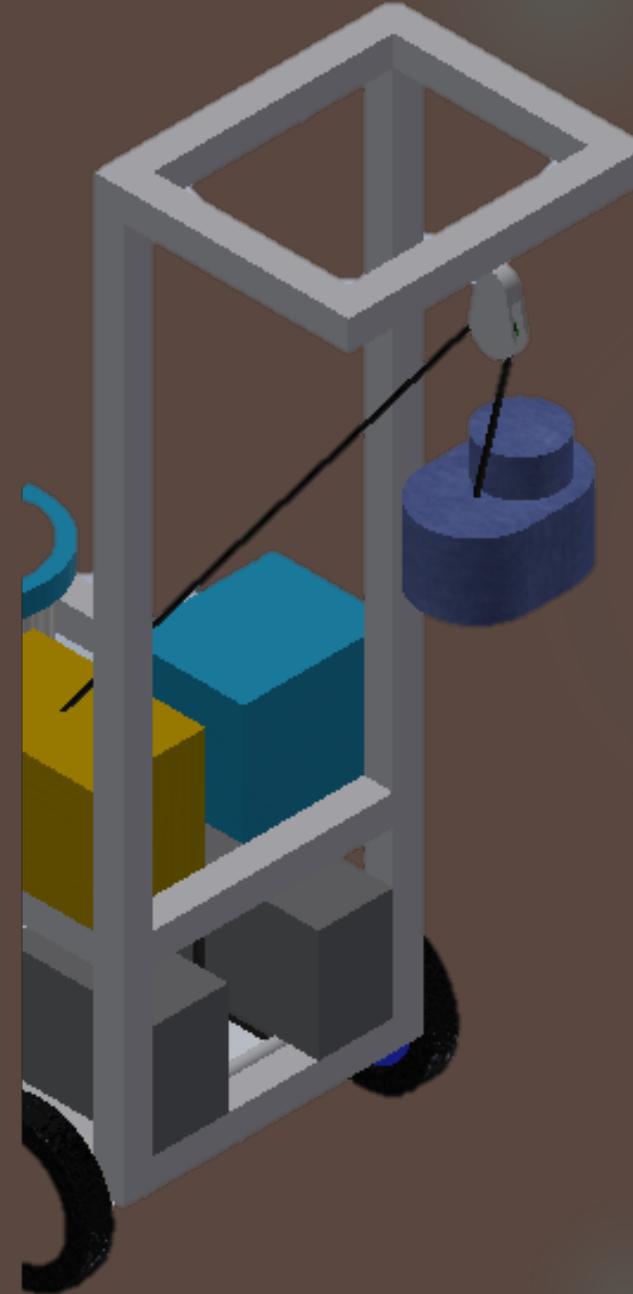
Pomp

- ▶ Bilgenpomp
- ▶ 63L/min
- ▶ 12V op 3,75A (max)



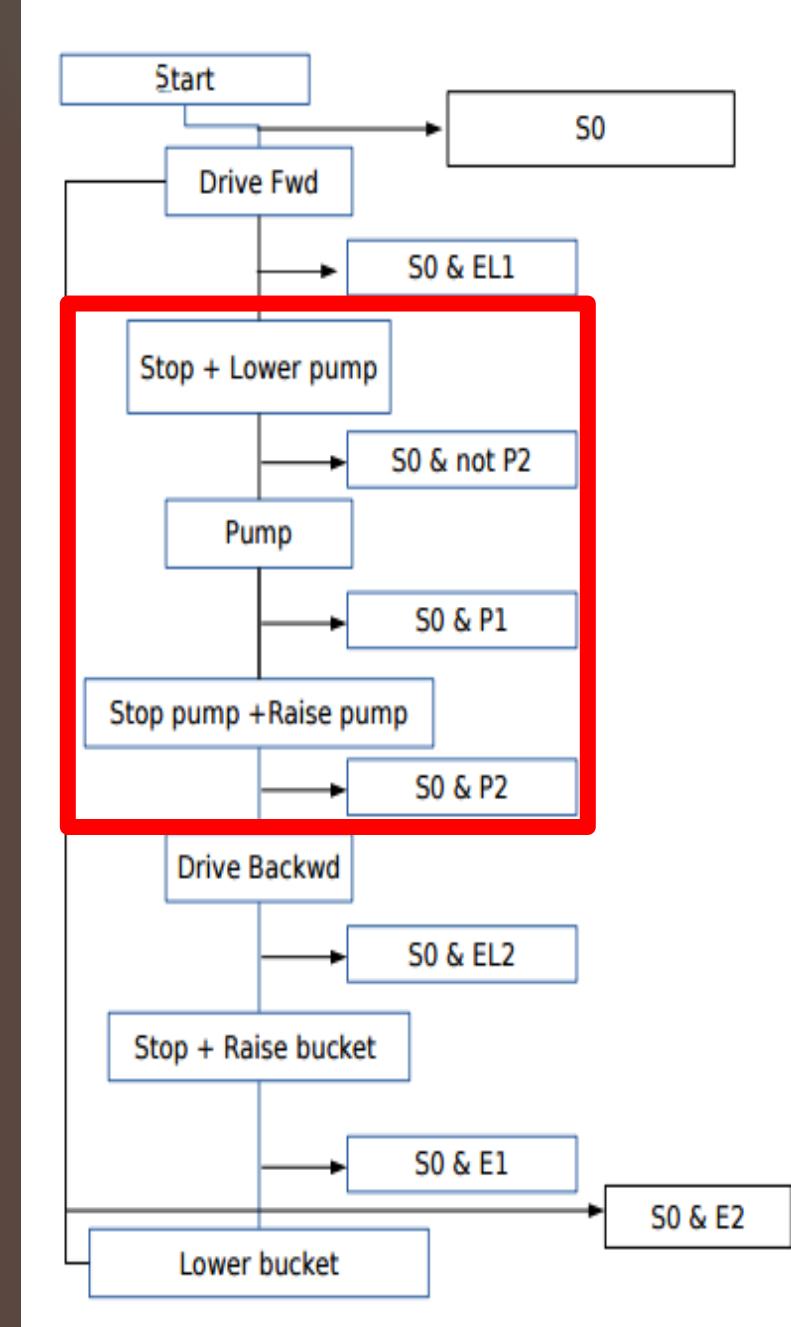
Pomp

- ▶ Bilgenpomp
- ▶ 63L/min
- ▶ 12V op 3,75A (max)
- ▶ Plaatsing op frame



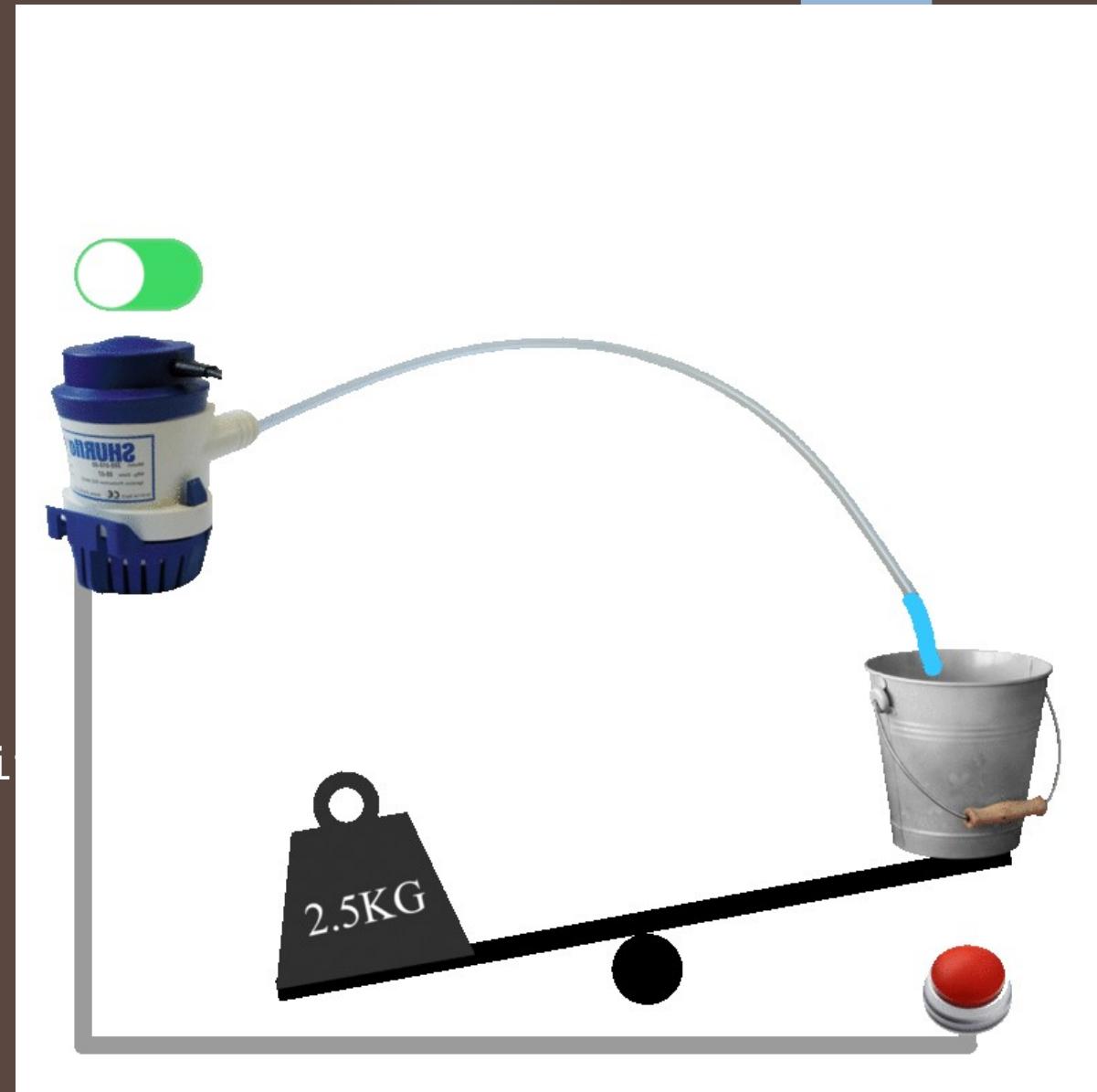
Pomp

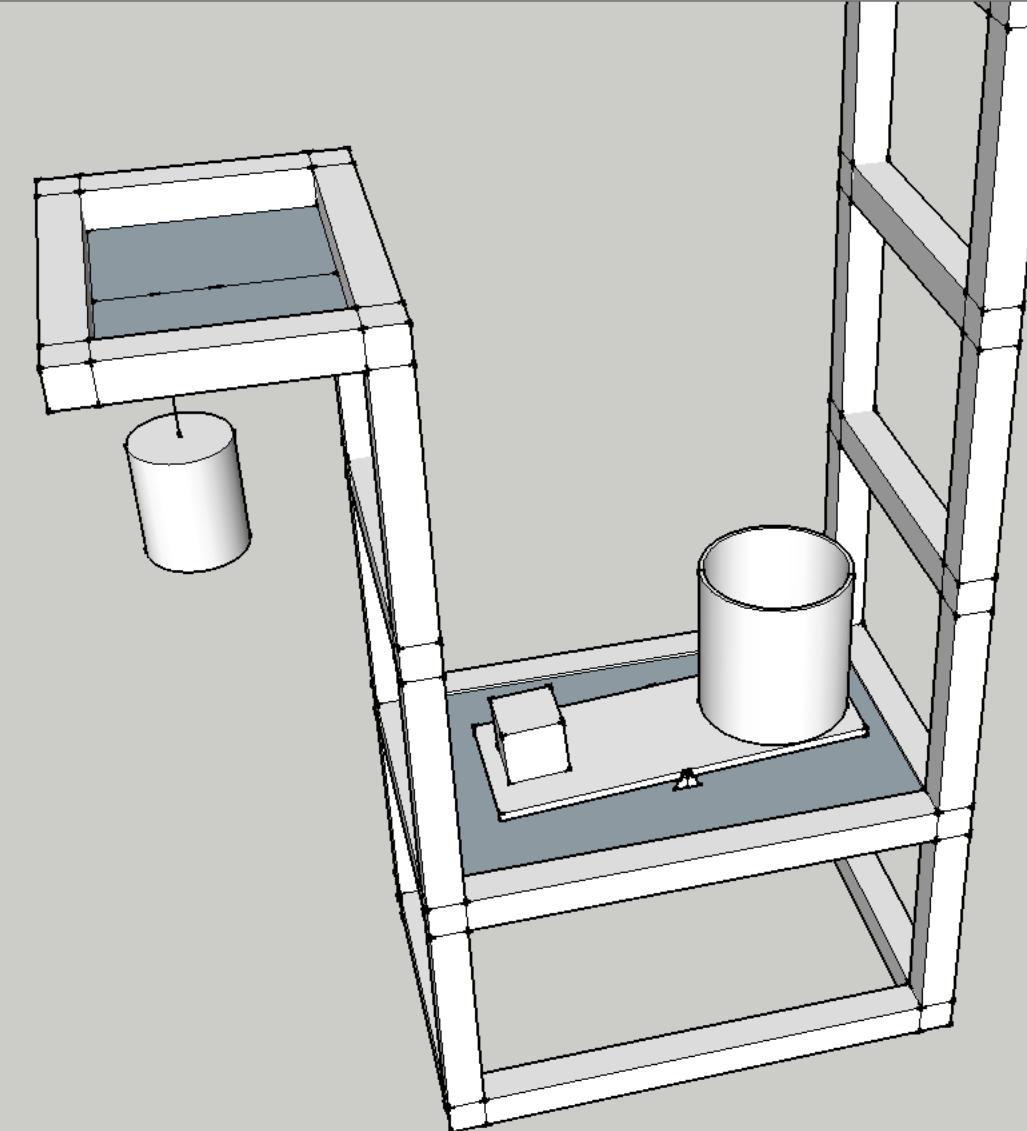
- ▶ Bilgenpomp
- ▶ 63L/min
- ▶ 12V op 3,75A (max)
- ▶ Plaatsing op frame
- ▶ Verandering van staat



Pomp

- ▶ Bilgenpomp
- ▶ 63L/min
- ▶ 12V op 3,75A (max)
- ▶ Plaatsing op frame
- ▶ Verandering van staat
 - ▶ Wanneer emmer vol genoeg is: swi





Pomp

- ▶ Bilgenpomp
- ▶ 63L/min
- ▶ 12V op 3,75A (max)
- ▶ Plaatsing op frame
- ▶ Verandering van staat
 - ▶ Wanneer emmer vol genoeg is: switch
- ▶ Video over werking





Emmer

Emmer

- ▶ Gebogen rail

Emmer

- ▶ Gebogen rail
- ▶ Karretje met assen en kogellagers geklemd rond rail

Emmer

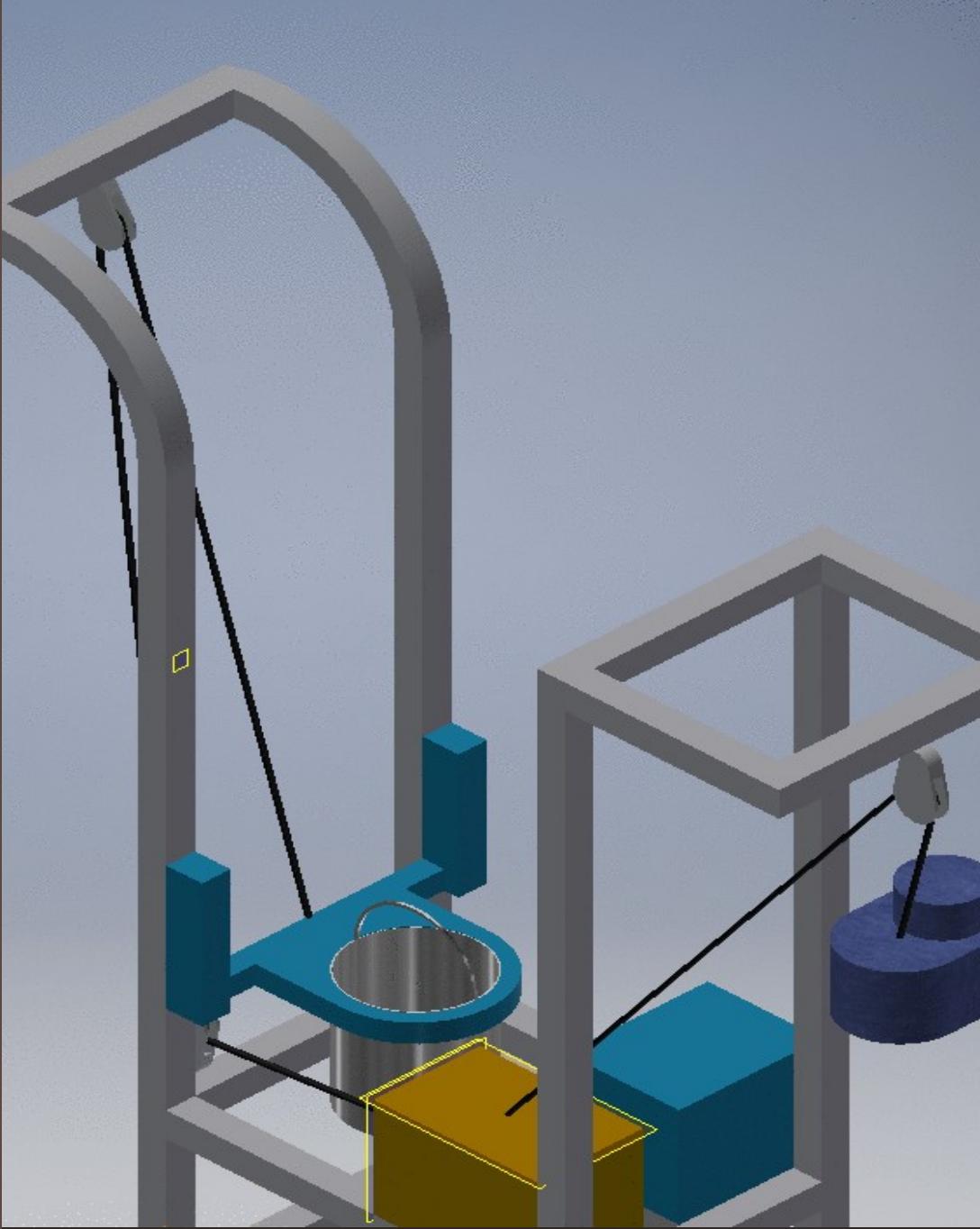
- ▶ Gebogen rail
- ▶ Karretje met assen en kogellagers geklemd
- ▶ Zware motor trekt gewicht snel omhoog



Emmer

- ▶ Gebogen rail
- ▶ Karretje met assen en kogellagers geklemd
- ▶ Zware motor trekt gewicht snel omhoog
- ▶ Trial & error goede buiging

Beweging emmer



Documentatie



No description, website, or topics provided.

131 commits

5 branches

0 releases

6 contributors

MIT

Branch: master ▾

New pull request

Create new file

Upload files

Find file

Clone or download ▾

 themaawaa	updated presentatie	Latest commit 19463ed 20 minutes ago
 .github	Issue template	a month ago
 3D_MODELS	gif animation of frame with dimensions	3 days ago
 DRIVING	Merge branch 'master' of github:ccmteamap/robot	3 days ago
 EMPTY_BUCKET	Description empty bucket added	5 months ago
 GIT_MANUAL	INSTALL.md upgedate met ssh instructies	5 months ago
 OFFERTES	spelfoutje	a month ago
 PUMPING_WATER	werkend motorcontroller schema	10 days ago
 docs	updated presentatie	20 minutes ago
 LICENCE.md	capitalize + Autheur	5 months ago
 README.md	Update README.md	a month ago

9/03 - 16/03	17/03 - 24/03	25/03 - 1/04	2/04 - 9/04	10/04 - 17/04
Sponsoring CCM	Bouwen test parcours	Afhebben frame	Afhebben emmer	Video rijdende robot 12/04
Bestellen onderdelen	Bouwen frame	Starten besturing emmer	Starten besturing pomp	Afhebben besturing
		Bug fixing	Optimalisatie	Optimaliseren besturing
		Eventuele componenten bijbestellen	Bug fixing	Optimaliseren emmer
		Starten besturing robot		
18/04 - 25/04	26/04-3/05	4/05-11/5	12/5 - 17/5	Goal / deadline
			17/05 ccm wedstrijd	Deadline CCM
			Debugging & testing	Nog niet gestart
			Improvements, bug fixing	Al gestart
				Bug fixing & optimalisatie

PLANNING

Backlog	11		
#4 opened by RuvenSalamon driving			
① [EPIC] As a team we want to build a solution to move the robot back and forth #21 opened by koenschauwaert			
① As a team we want to program the software for Arduino to read-out sensors and start motors etc. #31 opened by koenschauwaert driving			
① [EPIC] As a customer I want a fully functioning robot #24 opened by koenschauwaert			
① As a customer I want my robot to have a professional appearance. #33 opened by RuvenSalamon general			
TODO	10		
① As a customer I want my robot to be able to deposit water into a water collection tower. #12 opened by RuvenSalamon bucket			
① As a customer want a robot that can drive/move forwards and backwards #2 opened by RuvenSalamon driving			
① As a team we want the robot to drive at least 5 times back and forth #20 opened by koenschauwaert driving			
① As a customer I want the robot to extract water from the basin at a min. height of 70cm #13 opened by koenschauwaert pump			
In progress	2		
① As a customer I want to receive a detailed presentation #28 opened by koenschauwaert			
① As a customer I want my robot to not deviate from the route and keep driving straight #3 opened by RuvenSalamon driving			
Nearly there	3		
① As a team we want to imitate the build of the track, so we can practice on our own. #34 opened by koenschauwaert general			
① As a customer I want a clear overview of a prototype in Inventor which shows the operation #6 opened by koenschauwaert			
① As a programmer I want my robot to have wireless communication so I can have real time debugging information. #35 opened by RuvenSalamon general			
Done	8		
① As a team we want to program the software for Arduino to read-out sensors and start motors etc. #29 opened by koenschauwaert driving pump			
① As a customer we want the teams to use the bucket we provide with the dimensions (diameter x height) of 17cm x 16cm #19 opened by koenschauwaert bucket			
① As a team we want to program the software for Arduino to read-out sensors and start motors etc. #30 opened by koenschauwaert driving pump			
① As a customer I want a mechanical and easy to use			

SCRUM

Links

- ▶ Github: <https://github.com/ccmteamap/robot>
- ▶ Wiki: <https://github.com/ccmteamap/robot/wiki>
- ▶ Info CCM: <http://www.ccm.nl/nl/trofee/>
- ▶ Issues & sprints: <https://github.com/ccmteamap/robot/issues>
- ▶ Online scrum: <https://github.com/ccmteamap/robot/projects/1>