

Teamleden: Mike Meyers, Kay Elst en Bryan Lameir

Tussentijdse Portfolio Smart Systems

Inhoud

TaakVerdeling.....	2
Doel	2
Systeemspecificaties	2
Scrumplan.....	3
Geplande Aankopen	4
Planning.....	4
Voortgangspresentaties + samenvatting van de feedback	5
Handleiding(en) bij het uitgewerkte product	5
Literatuurlijst + wat gebruikt/gelezen	5
Lijst met relevante URL's + waarom relevant	6
Alle code van ontwikkelde testsoftware (+ commentaar!)	7
Eigen relevante toevoegingen	10
Relevante datasheets	10

Projectplan Smart Systems

TaakVerdeling

Software: Kay Elst

Hardware: Mike Meyers

PCB en Matrix: Bryan Lameir

Doel

In module 3 is het de bedoeling om een Robot Chassis RP5-CH02 te laten rijden. Dit wil zeggen dat de auto vooruit en achteruit kan rijden, alsook moet hij links en rechts kunnen draaien. Het moet ook kunnen afremmen en de nabijheid van voorwerpen kunnen detecteren. Het moet ook nog kunnen communiceren met een externe schakeling, namelijk laptop, microcontroller, ...

In module 4 is het de bedoeling dat het project van vorige module verder wordt uitgewerkt. Dus de voertuigen die we gebouwd hebben moeten autonoom rijden. Hiervoor gaan we gebruik maken van sensoren. Dit deel omvat 3 delen:

- In het eerste deel zal iedere groep zijn schakeling verder afwerken. De PCB's dienen bestukt en getest te worden.
- In het tweede deel is het de bedoeling dat jullie de voertuigen van de nodige sensoren voorzien zodat deze in staat zijn om hindernissen te detecteren. Het is hierbij ook de bedoeling dat jullie ervoor zorgen dat de voertuigen autonoom kunnen rijden.
- Er zullen ook nog een aantal programmeeropdrachten voor de microcontroller gegeven worden in oplopende moeilijkheidsgraad. Hierdoor zullen de voertuigen een steeds complexer parcours aankunnen. Deze moeten in volgorde worden afgewerkt.

Systeemspecificaties

PCB:

- Voor het aansturen van de motors
- Deze motors werken op 6V DC en 210mA
- Om objecten rond ons te detecteren gebruiken we ultrasoon sensors
- Communicatie met de laptop via een RF module
- Schakelaar voor de robot op en af te zetten.
- Afremmen van de robot

Arduino:

- Om de ultrasoon sensoren uit te lezen.
- Communiceren met de laptop via een RF module.
- Vooruit rijden
- Achteruit rijden
- De robot op en af kunnen zetten via de schakelaar
- Links draaien
- Rechts draaien
- Afremmen van de robot

Scrumplan

Smart Systems - Bryan Lameir; Mike Meyers, Kay Elst

Smart Systems

☆

Team Visible

Backlog

De auto moet objecten kunnen detecteren

De auto moet Een parcours kunnen afleggen

Add a card...

Sprint 1 ToDo 22/02 - 06/03

Add a card...

Sprint 1 Done 22/02 - 06/03

Schema opstellen voor aansturen motoren met fritzing

Mar 6

LB

De Componenten verzamelen

Mar 1

LB

Projectplan afwerken

1

MM

Beslissen welke componenten we willen gebruiken voor het autootje

Feb 28

MM

Code schrijven voor vooruit achteruit

Mar 6

KE

Een projectplan opstellen

Feb 28

MM

Schrijven van code voor links en rechts Draaien

Mar 6

KE

Research doen over de Componenten voor de PCB

1

1

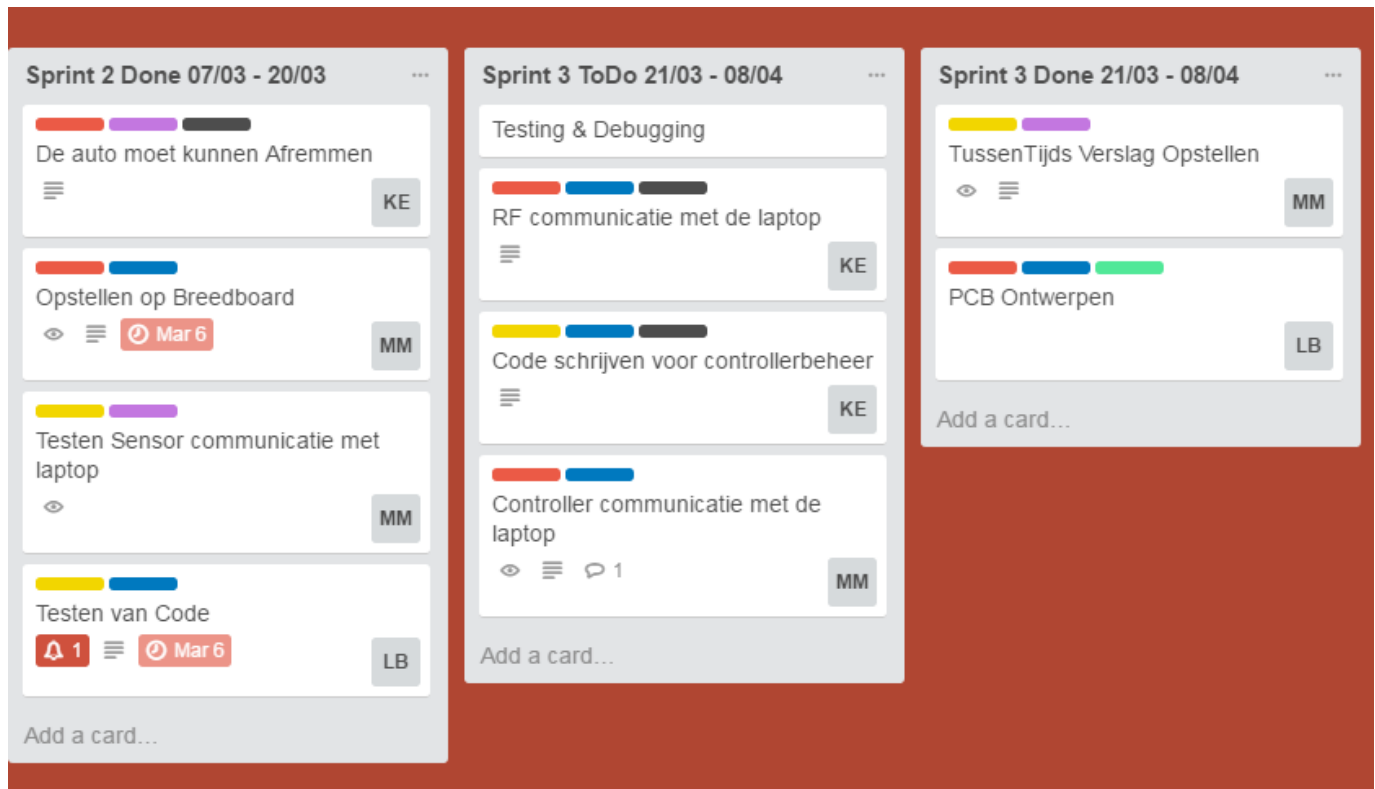
Feb 28

LB

Add a card...

Sprint 2 ToDo 07/03 - 20/03

Add a card...



Geplande Aankopen

- 2 x IC met h-brug (SN754410)
- ultrasoon sensor (3)
- transmitter & receiver
- oplaadbare batterijen of normale AA batterijen
- schakelaar

Planning

- Sprint 1 22/02-06/03
 - ✓ Schema opstellen voor aansturen motoren
 - ✓ Research componenten PCB
 - ✓ Beslissen welke componenten we gebruiken
 - ✓ Componenten verzamelen
 - ✓ Code voor vooruit en achteruit
 - ✓ Projectplan opstellen
 - ✓ Code voor links en rechts draaien
 - ✓ Trelloboard aanmaken
 - ✓ Taakverdeling maken
 - ✓ Projectplan afwerken
- Sprint 2 07/03-20/03
 - ✓ Code voor afremmen
 - ✓ Opstellen op breedboard
 - ✓ Code testing
 - ✓ Testen sensor communicatie met laptop

- Sprint 3 21/03-08/04
 - ✓ Testing en debugging
 - ✓ Code voor Controllerbeheer
 - ✓ Controller communicatie met laptop
 - ✓ RF communicatie met laptop
 - ✓ Tussentijdsverslag opstellen
 - ✓ PCB ontwerpen

Voortgangspresentaties + samenvatting van de feedback

We hebben in lesweek 3 een presentatie moeten geven van ons projectplan. Hier hebben de docenten dan feedback gegeven die we in ons project gaan gebruiken. De feedback was dat onze schakeling werkte, de auto kon rijden. Dus dat we dan de schakeling van ons breadboard omzetten naar een matrix zodat de componenten ook veilig zijn bij vervoer. En het is ook praktischer. Ons scrum plan was goed, alleen zeiden ze dat onze datums verkeerd waren van de sprints, dus dit hebben we dan ook onmiddellijk aangepast in trello. En ook nog bij ons scrumplan moesten we sprints van volgende module al meer uitwerken, zodat we al een goed overzicht hebben wanneer we aan de volgende sprints beginnen.

Handleiding(en) bij het uitgewerkte product

Onze auto werkt op batterijen, dus als ze moesten leeg zijn kan u deze best gewoon vervangen. We maken ook gebruik van sensoren zodat de auto en diegene die het bestuurt weet wanneer er een object in de buurt is. Voor de besturing van de auto maken we gebruik van de pijltjes van het toetsenbord op de laptop. Deze zijn:

- Vooruit: pijl naar boven
- Achteruit: pijl naar onder
- Naar links draaien: pijl naar links
- Naar rechts draaien: pijl naar rechts

Literatuurlijst + wat gebruikt/gelezen

Voor de volgende onderwerpen hebben we telkens de sites erbij gezet die we gebruikten of gelezen hebben om meer informatie te hebben over de componenten of de code die we daarvoor gaan schrijven.

Ultrasonic sensor:

<http://www.instructables.com/id/Ultrasonic-Range-detector-using-Arduino-and-the-SR/>

<http://playground.arduino.cc/Main/UltrasonicSensor>

<http://www.tautvidas.com/blog/2012/08/distance-sensing-with-ultrasonic-sensor-and-arduino/>

rf Module:

<http://arduinobasics.blogspot.be/2014/06/433-mhz-rf-module-with-arduino-tutorial.html>

http://www.pjrc.com/teensy/td_libs_VirtualWire.html

H-Brug:

<http://html.alldatasheet.com/html-pdf/28615/TI/SN754410/22/1/SN754410.html>

Robot Chassis:

<https://www.pololu.com/product/1060>

Besturing via PC:

https://processing.org/reference/keyPressed_.html

<https://processing.org/reference/keyCode.html>

<https://learn.sparkfun.com/tutorials/connecting-arduino-to-processing>

<http://www.instructables.com/id/Moving-a-Motor-Using-Arduino-and-Serial-Communcati/?ALLSTEPS>

Lijst met relevante URL's + waarom relevant

Voor de ultrasone sensor was deze site zeer relevant, omdat we aan de hand hiervan onze sensoren even hebben kunnen testen en om te zien wa de waardes zijn die ze teruggeven. Deze site liet ons ook zien hoe we ze moeten aansluiten.

<http://www.tautvidas.com/blog/2012/08/distance-sensing-with-ultrasonic-sensor-and-arduino/>

Voor de H-brug was deze site zeer relevant, omdat deze uitlegt wat het component juist is en hoe we deze moeten aansluiten. Het gaf ook een inwendig schema mee van de component wat altijd handig is.

<http://html.alldatasheet.com/html-pdf/28615/TI/SN754410/22/1/SN754410.html>

Voor de besturing via de PC waren al deze sites relevant, omdat bij de keypressed en keycode de sites lieten zien hoe je kan onthouden welke key is ingedrukt, wat we wel moeten weten doordat onze besturing met de pijltjes is. De keycode hebben we gebruikt om speciale keys te kunnen detecteren zoals de pijltjes toetsen. Dan hebben we nog 2 sites en die hebben we vooral gebruikt om te zien hoe dat je juist een connectie maakt van de arduino naar de motor en ook nog de serial communicatie.

https://processing.org/reference/keyPressed_.html

<https://processing.org/reference/keyCode.html>

<https://learn.sparkfun.com/tutorials/connecting-arduino-to-processing>

<http://www.instructables.com/id/Moving-a-Motor-Using-Arduino-and-Serial-Communcati/?ALLSTEPS>

Alle code van ontwikkelde testsoftware (+ commentaar!)

Code voor vooruit, achteruit, links en rechts, remmen

```
//Libraries Aanspreken
#include <VirtualWire.h>

//INTS aanmaken
int LinksVooruit = 6;
int LinksAchteruit = 7;
int RechtsVooruit = 4;
int RechtsAchteruit = 3;
int EnableRechts = 2;
int EnableLinks = 5;
int Work = false;

void setup() {
  // put your setup code here, to run once:
  //Zet alle pins als Output
  pinMode(LinksVooruit, OUTPUT);
  pinMode(LinksAchteruit, OUTPUT);
  pinMode(RechtsVooruit, OUTPUT);
  pinMode(RechtsAchteruit, OUTPUT);
  pinMode(EnableLinks, OUTPUT);
  pinMode(EnableRechts, OUTPUT);
  //zorg ervoor dat seriële communicatie enabled is voor de input.
  Serial.begin(9600);
}

void loop() {
  // put your main code here, to run repeatedly:
  //FIRST TEST
  Vooruit(3000);
  DraaiRechts(3000);
  DraaiLinks(3000);
  Achteruit(3000);
  Stop(10000);
}

void Drive(int LeftWheel, int RightWheel)
{
  switch(LeftWheel)
  {
    //links wiel vooruit
    case 1:
      digitalWrite(EnableLinks, HIGH);
      digitalWrite(LinksVooruit, HIGH);
      digitalWrite(LinksAchteruit, LOW);
      break;
    //links wiel remt
    case 0:
```



```

    digitalWrite(EnableLinks, LOW);
    digitalWrite(LinksVooruit, LOW);
    digitalWrite(LinksAchteruit, LOW);
    break;
case -1:
    digitalWrite(EnableLinks, HIGH);
    digitalWrite(LinksVooruit, LOW);
    digitalWrite(LinksAchteruit, HIGH);
    break;
}
switch(RightWheel)
{
    case 1:
        digitalWrite(EnableRechts, HIGH);
        digitalWrite(RechtsVooruit, HIGH);
        digitalWrite(RechtsAchteruit, LOW);
        break;
        //links wiel remt
    case 0:
        digitalWrite(EnableRechts, LOW);
        digitalWrite(RechtsVooruit, LOW);
        digitalWrite(RechtsAchteruit, LOW);
        break;
    case -1:
        digitalWrite(EnableRechts, HIGH);
        digitalWrite(RechtsVooruit, LOW);
        digitalWrite(RechtsAchteruit, HIGH);
        break;
}
}

void CheckSchakelaar()
{
}

void Vooruit(int delayVal)
{
    Drive(1,1);
    delay(delayVal);
    Stop(1000);
}

void DraaiRechts(int delayVal)
{
    Drive(-1, 1);
    delay(delayVal);
    Stop(1000);
}

void DraaiLinks(int delayVal)

```

```

{
  Drive(1, -1);
  delay(delayVal);
  Stop(1000);
}

void Achteruit(int delayVal)
{
  Drive(-1,-1);
  delay(delayVal);
  Stop(1000);
}

void Stop(int delayVal)
{
  Drive(0,0);
  delay(delayVal);
}

```

Code voor sensoren

```

const int trigPin = 2;
const int echoPin = 4;

void setup() {
  // initialize serial communication:
  Serial.begin(9600);
}

void loop()
{
  // establish variables for duration of the ping,
  // and the distance result in inches and centimeters:
  long duration, inches, cm;

  // The sensor is triggered by a HIGH pulse of 10 or more microseconds.
  // Give a short LOW pulse beforehand to ensure a clean HIGH pulse:
  pinMode(trigPin, OUTPUT);
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  // Read the signal from the sensor: a HIGH pulse whose
  // duration is the time (in microseconds) from the sending
  // of the ping to the reception of its echo off of an object.
  pinMode(echoPin, INPUT);
  duration = pulseIn(echoPin, HIGH);
}

```

```

// convert the time into a distance
inches = microsecondsToInches(duration);
cm = microsecondsToCentimeters(duration);

Serial.print(inches);
Serial.print("in, ");
Serial.print(cm);
Serial.print("cm");
Serial.println();

delay(100);
}

long microsecondsToInches(long microseconds)
{
  // According to Parallax's datasheet for the PING))) , there are
  // 73.746 microseconds per inch (i.e. sound travels at 1130 feet per
  // second). This gives the distance travelled by the ping, outbound
  // and return, so we divide by 2 to get the distance of the obstacle.
  // See: http://www.parallax.com/dl/docs/prod/acc/28015-PING-v1.3.pdf
  return microseconds / 74 / 2;
}

long microsecondsToCentimeters(long microseconds)
{
  // The speed of sound is 340 m/s or 29 microseconds per centimeter.
  // The ping travels out and back, so to find the distance of the
  // object we take half of the distance travelled.
  return microseconds / 29 / 2;
}

```

Code voor communicatie

We hebben nog geen code voor Bluetooth.

We hadden het eerst geprobeerd met RF module maar deze kon alleen maar frequenties oppikken en niets doorsturen. Dus we hadden wel al code voor deze componenten voorzien, maar we kregen ze niet meer werkend. Nu zijn we overgeschakeld naar het andere en daarvoor gaan we in de vakantie code voor voorzien, zodat we tegen de presentatie voorbrenging het kunnen laten zien in de demo.

Eigen relevante toevoegingen

We hebben tot hiertoe nog geen eigen relevante toevoegingen voorzien.

Relevante datasheets

Voor de datasheets hebben we vooral gebruik gemaakt van deze:

Die van de sensoren en die van de H-brug, omdat we moesten weten hoe we deze gingen aansluiten.

Voor verdere uitleg, zie in bijlagen.