

EIND OPLEVERING

OIT11 2018

Naam: Bram van Gils
PCN: 401949

Contents

Module 1.1	2
Opdracht 4.2	2
Opdracht 4.3	3
Module 2.1	5
Flow-Chart.....	5
Code	5
Module 3.1	8
Code	8
Module 3.2 – 3.5	9
Flow-Chart.....	9
Code:	10
Variabelen en CheckPaused Method.....	10
CalibrateValues Method	10
PowerMotors + StopMotors Method	10
Evade Method.....	11
Main Method	11
Bronnen.....	12

Module 1.1

```
task main()
{
    while(getTouchValue(S1) != 1){}; // wacht tot touch sensor wordt ingedrukt.

    for(byte x = 0; x < 4; ++x) // herhaal onderstaande sequentie 4 maal.
    {
        setMotorSyncEncoder(motorB, motorC, 0, 1080, 100); // rijd 50 cm naar voren.
        waitUntilMotorStop(motorB);
        setMotorSyncEncoder(motorB, motorC, 100, 180, 100); // maak kwartslag draai.
        waitUntilMotorStop(motorB);
    }
}
```

Module 1.2

```
task main()
{
    while(getTouchValue(S1) != 1){}; // wacht tot touch sensor wordt ingedrukt.

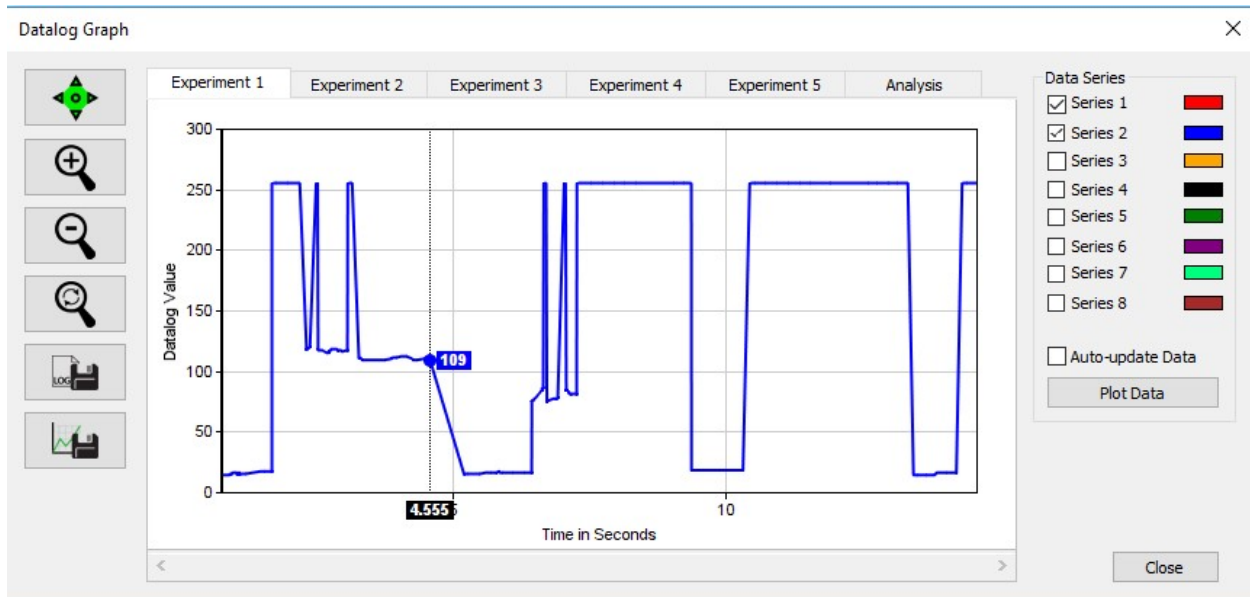
    for(byte x = 0; x < 8; ++x) // herhaal onderstaande sequentie 8 maal.
    {
        setMotorSyncEncoder(motorB, motorC, 0, 1080, 100); // rijd 50 cm naar voren.
        waitUntilMotorStop(motorB);
        setMotorSyncEncoder(motorB, motorC, 100, 180, 100); // maak kwartslag draai.
        waitUntilMotorStop(motorB);
    }
}
```

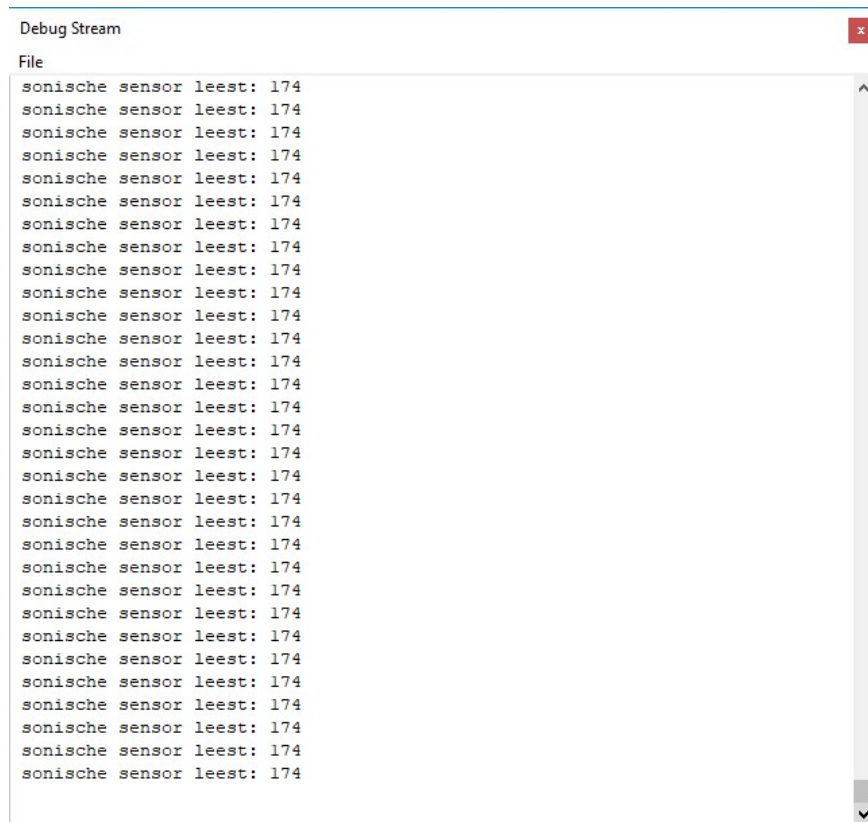
Module 1.3

```
task main()
{
    resetGyro(S2); // zet de gyro terug op 0
    while(getTouchValue(S1) != 1){}; // wacht tot touch sensor wordt ingedrukt.

    while(getGyroDegrees(S2) <= 360) // herhaal onderstaande actie totdat een volle draai gemaakt wordt.
    {
        datalogAddValueWithTimeStamp(1, getUSDistance(S4)); // voeg sonische metingen toe aan datalog.
        writeDebugStreamLine("sonische sensor leest: %d", getUSDistance(S4)); // voeg sonische metingen toe aan debugstream.

        motor(motorB) = 5;
        motor(motorC) = -5; // draai met de klok mee
    }
}
```

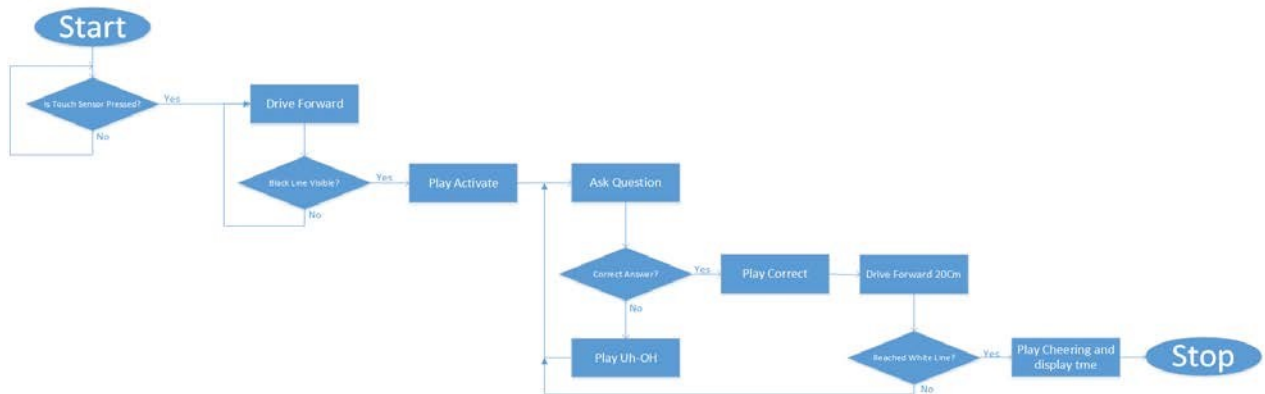




(Resultaten lijken op elkaar omdat de sensor 16.610 metingen heeft gemaakt)

Module 2.1

Flow-Chart



Link voor betere weergave <https://drive.google.com/open?id=1B-IYXPB1uSaZpWDp1falolwQOYqI-Dz8>

Code

```
1  bool waitanswer = false;
2  bool finished = false;
3  float time;
4  bool correctleft = true;
5  string options;
6
7  task main()
8  {
9      //wacht tot de touch sensor wordt ingedrukt
10     while(getTouchValue(S1) != 1){};
11     clearTimer(T1);
12
13     //rijd naar voren totdat de zwarte lijn gevonden wordt
14     while(getColorReflected(S3) > 7)
15     {
16         setMotor(motorB, 50);
17         setMotor(motorC, 50);
18     }
19
20     //stop en play Activate
21     setMotor(motorB, 0);
22     setMotor(motorC, 0);
23     playSoundFile("Activate");
```

```

25 while(!finished) //herhaal zolang de witte lijn nog niet gevonden is
26 {
27     //pak random random waarden en bereken antwoord
28     int value1 = 1 + random(9);
29     int value2 = 1 + random(9);
30     int answer = value1 * value2;
31
32     //display vraag
33     string question;
34     stringFormat(question, "wat is %d X %d?", value1, value2);
35     displayBigTextLine(4, question);
36
37     //voeg mutatie toe aan een van de waarden en maak een fout antwoord
38     switch(random(3))
39     {
40     case 0:
41         value1++;
42         break;
43
44     case 1:
45         value1--;
46         break;
47
48     case 2:
49         value2++;
50         break;
51
52     case 3:
53         value2--;
54         break;
55     }
56     int notanswer = value1 * value2;
57
58     //display antwoorden
59     switch(random(1))
60     {
61     case 0:
62         stringFormat(options, " %d %d ", answer, notanswer);
63         correctleft = true;
64         break;
65
66     case 1:
67         stringFormat(options, " %d %d ", notanswer, answer);
68         correctleft = false;
69         break;
70     }
71     displayBigTextLine(10, options);

```

```

72
73 //wacht tot knop wordt ingedrukt
74 waitUntil(getButtonPress(buttonLeft) || getButtonPress(buttonRight));
75 waitanswer = true;
76 while(waitanswer == true)
77 {
78
79 //als het antwoord fout is wacht 0,2 seconden play "Uh-oh" en verlaat loop
80 if((getButtonPress(buttonLeft) && !correctleft) || (getButtonPress(buttonRight) && !correctleft))
81 {
82     playSoundFile("Uh-oh");
83     delay(200);
84     waitanswer = false;
85 }
86
87 //als het antwoord goed is:
88 else if(!getButtonPress(buttonLeft) && correctleft) || (getButtonPress(buttonRight) && !correctleft)
89 {
90     //play good en start motor
91     playSoundFile("Good");
92     setMotor(motorB, 50);
93     setMotor(motorC, 50);
94
95 //check gedurende 1 seconde of de grond wit is
96 clearTimer(T2);
97 while(getTimerValue(T2) < 1000)
98 {
99     //zo ja stop motor en stop vragen stellen
100     if(getColorReflected(S3) > 40)
101     {
102         finished = true;
103
104         setMotor(motorB, 0);
105         setMotor(motorC, 0);
106     }
107 }
108
109 //na 1 sec stop motoren wacht op nieuwe vraag
110 setMotor(motorB, 0);
111 setMotor(motorC, 0);
112 waitanswer = false;
113 }
114 }
115
116 //play cheer
117 playSoundFile("Cheering");
118
119 //clear display
120 displayBigTextLine(10, " ");
121 displayBigTextLine(4, " ");
122
123 //bereken tijd in seconden
124 time = getTimerValue(T1) / 1000;
125
126 //display eindtijd
127 displayBigTextLine(6, "eind tijd:");
128 displayBigTextLine(8, "%d sec", time);
129
130 wait(3000);
131 }
132

```


Module 3.1

Code

```
while (true)
{
    // Read input message.
    // readMailboxIn() is non-blocking and returns "" if there is no message.
    readMailboxIn("EV3_INBOX0", msgBufIn);
    if (strcmp(msgBufIn, "") != 0)
    {
        if(strcmp(msgBufIn, "Forward") == 0)
        {
            displayBigTextLine(8, "Forward");
            setMotor(motorB, 50);
            setMotor(motorC, 50);
        }

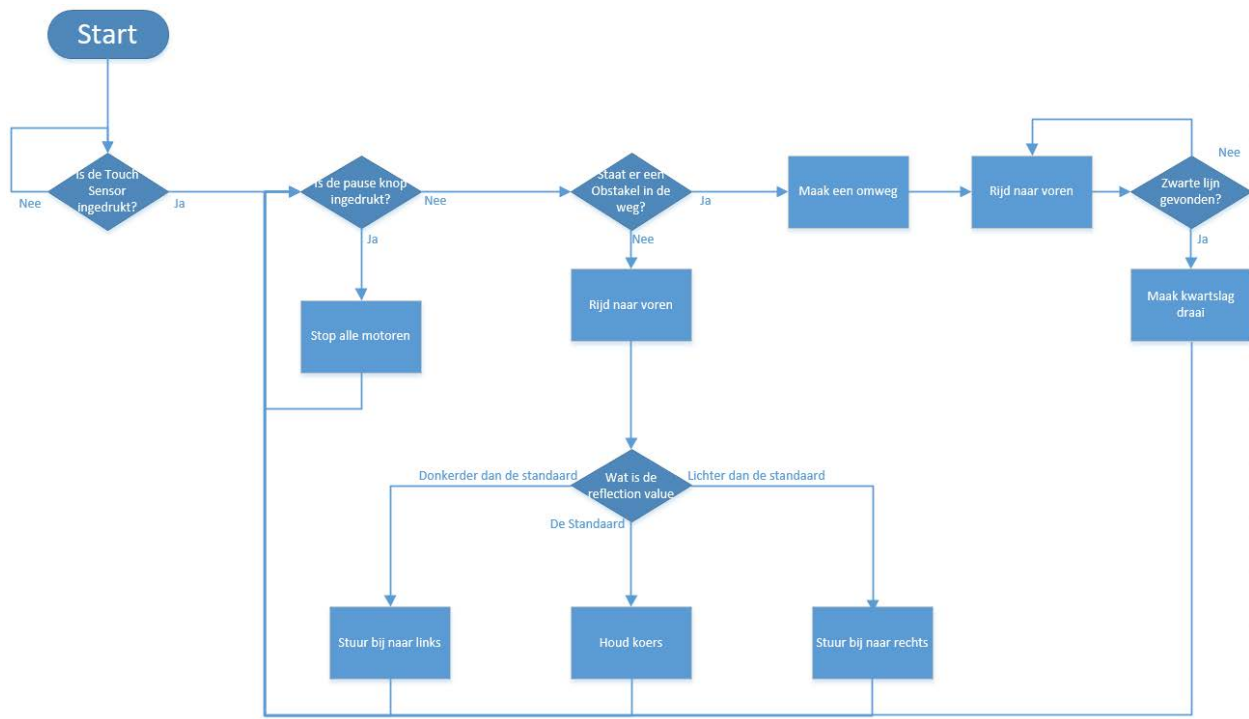
        if(strcmp(msgBufIn, "Backward") == 0)
        {
            displayBigTextLine(8, "Backward");
            setMotor(motorB, -50);
            setMotor(motorC, -50);
        }

        if(strcmp(msgBufIn, "Left") == 0)
        {
            displayBigTextLine(8, "left");
            setMotor(motorB, 0);
            setMotor(motorC, 50);
        }

        if(strcmp(msgBufIn, "Right") == 0)
        {
            displayBigTextLine(8, "Right");
            setMotor(motorB, 50);
            setMotor(motorC, 0);
        }
    }
}
```

Module 3.2 – 3.5

Flow-Chart



Link voor betere Weergave:

<https://drive.google.com/open?id=1mMaBmnQnCb5VIEaaypnY8kOFj9mnrW-c>

Code:

Variabelen en CheckPaused Method

```
int DarkestValue = getColorReflected(S3);
int LightestValue = getColorReflected(S3);
int StandardLightValue;
int MaxEnginePower = 10;
int Offset;
int HalfRange;

bool ButtonState;
bool PauseState = false;
|
void CheckPaused()
{
    if(getButtonPress(buttonEnter) && !ButtonState) // check of er een verandering is in de button state
    {
        PauseState = !PauseState;
    }

    ButtonState = getButtonPress(buttonEnter);
}
```

CalibrateValues Method

```
void CalibrateValues()
{
    if (DarkestValue > getColorReflected(S3)) // stel de donkerste waarde bij
    {DarkestValue = getColorReflected(S3);}

    if (LightestValue < getColorReflected(S3)) // stel de lichtste waarde bij
    {LightestValue = getColorReflected(S3);}

    StandardLightValue = (LightestValue + DarkestValue) / 2; // vind de algemene waarde die gevolgd wordt
    HalfRange = (LightestValue - DarkestValue) / 2; // vind de afwijking range
    Offset = StandardLightValue - getColorReflected(S3); // vind de mate van afwijking

    displayBigTextLine(8,"Offset = %d", Offset); // debug
}
```

PowerMotors + StopMotors Method

```
void PowerMotors()
{
    CalibrateValues();

    if(Offset < 0)
    {
        setMotor(motorC,(1+Offset / HalfRange) * MaxEnginePower); //Stuur naar rechts
        setMotor(motorB,MaxEnginePower);
    }
    else if(Offset > 0)
    {
        setMotor(motorC,MaxEnginePower);
        setMotor(motorB,(1-Offset /HalfRange) * MaxEnginePower); //Stuur naar links
    }
    else
    {
        setMotor(motorB,MaxEnginePower);
        setMotor(motorC,MaxEnginePower); // recht vooruit
    }
}

void MotorStop()
{
    setMotor(motorB, 0);
    setMotor(motorC, 0);
}
```

Evade Method

```
void Evade()
{
    setMotorSyncEncoder(motorB, motorC, -100, 180, 20); // maak kwartslag draai.
    waitUntilMotorStop(motorB);
    setMotorSyncEncoder(motorB, motorC, 0, 540, 20); // rijd 50 cm naar voren.
    waitUntilMotorStop(motorB);
    setMotorSyncEncoder(motorB, motorC, 100, 180, 20); // maak kwartslag draai.
    waitUntilMotorStop(motorB);
    setMotorSyncEncoder(motorB, motorC, 0, 540, 20); // rijd 50 cm naar voren.
    waitUntilMotorStop(motorB);
    setMotorSyncEncoder(motorB, motorC, 100, 180, 20); // maak kwartslag graden draai.
    waitUntilMotorStop(motorB);

    setMotor(motorB, 20);
    setMotor(motorC, 20);

    while(getColorReflected(S3) >= StandardLightValue) // zoekzwarte lijn
    {}

    setMotorSyncEncoder(motorB, motorC, -100, 180, 20); // maak kwartslag draai.
    waitUntilMotorStop(motorB);
}
```

Main Method

```
task main()
{
    while (getTouchValue(S1) == 0) {}

    while(true)
    {
        CheckPaused();

        if (!PauseState)
        {
            PowerMotors();
        }
        else
        {
            MotorStop();
        }

        if(getUSDistance(S4) < 20)
        {
            Evade();
        }

        delay(10);
    }
}
```

Bronnen

http://help.robotc.net/WebHelpMindstorms/Content/Resources/topics/LEGO_EV3/ROBOTC/Motors/setMotorSyncEncoder.htm

http://www.robotc.net/wikiarchive/Tutorials/Arduino_Projects/Mobile_Robotics/VEX/Using_encoders_to_drive_some_distance

http://help.robotc.net/Sandbox/Zendesk-Output/Content/Resources/topics/VEX_IQ/ROBOTC/Datalogging/datalogAddValueWithTimeStamp.htm

<https://stackoverflow.com/questions/822323/how-to-generate-a-random-number-in-c>

<https://stackoverflow.com/questions/2751406/while-with-multiple-conditions>

<https://sites.google.com/site/ev3basic/ev3-basic-programming/using-buttons-the-screen-and-theleds/lego-ev3-standard-bitmaps>

<https://msdn.microsoft.com/en-us/library/f355wky8.aspx>

<https://msdn.microsoft.com/en-us/library/f355wky8.aspx>

https://www.tutorialspoint.com/cprogramming/c_break_statement.htm

http://help.robotc.net/WebHelpVEX/index.htm#Resources/topics/ROBOTC_Debugger/Debug_Windows/Motors.htm

<https://www.youtube.com/watch?v=BjVeWRNiddE>

<https://stackoverflow.com/questions/16627588/what-is-the-correct-string-terminator-in-c>

https://www.tutorialspoint.com/cprogramming/c_arrays.htm

https://www.le.ac.uk/users/rjm1/cotter/page_22.htm

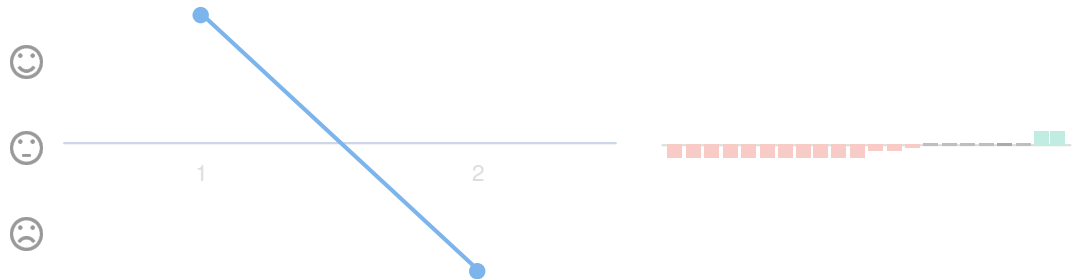
https://www.tutorialspoint.com/cprogramming/c_variables.htm

<https://www.geeksforgeeks.org/understanding-extern-keyword-in-c/>

<https://stackoverflow.com/questions/15707732/what-does-array-means>



Gils, Bram B.M.C.J. van ●



Checkpoint 3 Mar 15, 2018



Gils, Bram B.M.C.J. van

Mar 29, 2018

Feedback Module 3:

Algoritme is bijzonder goed bedacht.

Namen van alle methodes zijn duidelijk.

Alle methodes zijn goed verdeeld en erg overzichtelijk.

Format van de Main functie kan nog verbeterd worden.

Je bent ook de enige die de robot zelf laat bijsturen.

Feedback Presentatie:

Goed aantal Voorbeelden.

Erg levendige Opening.

De presentatie was vlot en zonder al teveel moeite

Maar IOT zat niet in het experiment.

Feedack Module 1:

Ik heb nooit feedback gekregen op module 1 omdat ik toen na de les doorgewerkt had.

Checkpoint 2 Mar 1, 2018



Marcelis, Ronald R. Mar 22, 2018

Heb ik sinds februari geen feedback meer gegeven?

Checkpoint 1 Feb 8, 2018



Gils, Bram B.M.C.J. van

Mar 1, 2018

Vergelijken met true of false hoeft niet.

Variabelen zo lokaal mogelijk definieren.

volgende keer methodes gebruiken.

erg netjes gecodeerd.

goede namen gebruikt.

ga zeker zo door.