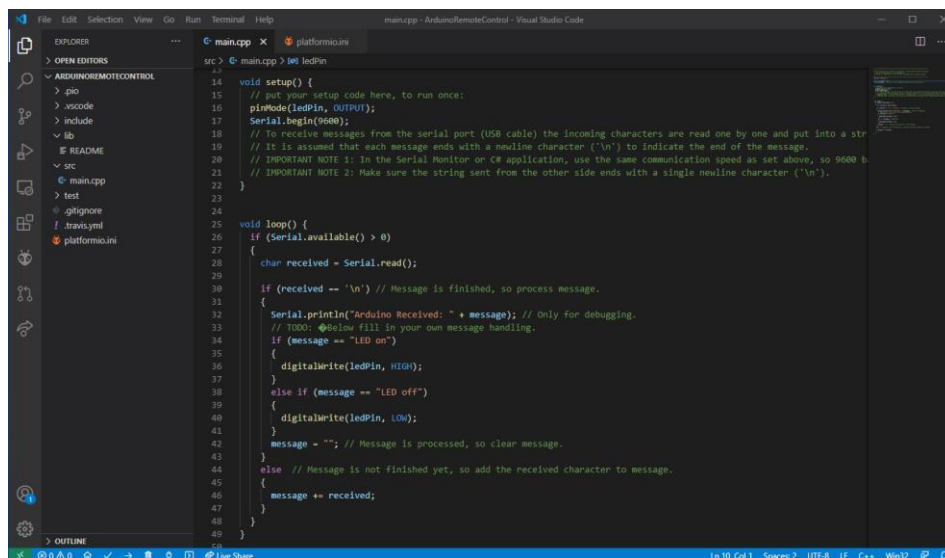
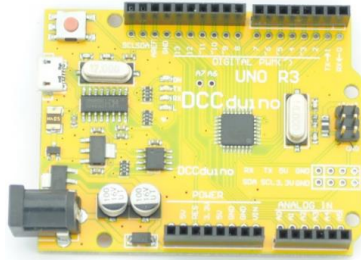


Werkdocument

Oriëntatie

A screenshot of the Visual Studio Code editor interface. The Explorer pane on the left shows a project named 'ARDUINOREMOTECONTROL' with files like 'main.cpp', 'platformio.ini', and 'test'. The main editor window displays the 'main.cpp' file, which contains C++ code for an Arduino. The code includes a setup function for serial communication and a loop function that reads serial data and controls an LED. The status bar at the bottom indicates the file is 'main.cpp' at line 10, column 1, using UTF-8 encoding and LF line endings.

```
14 void setup() {  
15     // put your setup code here, to run once:  
16     pinMode(ledPin, OUTPUT);  
17     Serial.begin(9600);  
18     // To receive messages from the serial port (USB cable) the incoming characters are read one by one and put into a str  
19     // It is assumed that each message ends with a newline character ('\n') to indicate the end of the message.  
20     // IMPORTANT NOTE 1: In the Serial Monitor or in application, use the same communication speed as set above, so 9600 b  
21     // IMPORTANT NOTE 2: Make sure the string sent from the other side ends with a single newline character ('\n').  
22 }  
23  
24  
25 void loop() {  
26     if (Serial.available() > 0)  
27     {  
28         char received = Serial.read();  
29  
30         if (received == '\n') // Message is finished, so process message.  
31         {  
32             Serial.println("Arduino Received: " + message); // Only for debugging.  
33             // TODO: Below fill in your own message handling.  
34             if (message == "LED on")  
35             {  
36                 digitalWrite(ledPin, HIGH);  
37             }  
38             else if (message == "LED off")  
39             {  
40                 digitalWrite(ledPin, LOW);  
41             }  
42             message = ""; // Message is processed, so clear message.  
43         }  
44         else // Message is not finished yet, so add the received character to message.  
45         {  
46             message += received;  
47         }  
48     }  
49 }
```

Version table

Version	Date	Description
1.0	20-11-2021	Initial version after review
1.1	12-07-2022	Added MiniBot project
1.2	26-07-2022	Added RobotGripper info

Contents

1	Inleiding.....	5
1.1	Aangeleverde bronnen.....	5
1.2	Demand based opzet	5
2	Installatie Arduino.....	7
2.1	Installatie Arduino IDE.....	7
2.2	Installatie Visual Studio Code met PlatformIO.....	7
2.3	Arduino Simulatie omgeving	7
3	Hello World! op Arduino.....	8
3.1	Challenges	8
3.1.1	Helloworld challenge	8
3.1.2	Serial.println challenge	9
4	Standaard programma structuur	10
4.1	Challenges	10
4.1.1	Program course challenge	10
4.1.2	Structure of Arduino sketch challenge	10
4.1.3	Function reference challenge	10
5	Hello World! op Arduino met Rich Shield.....	11
5.1	Challenges	11
5.1.1	RichShieldHelloWorld challenge.....	11
6	Output – aansturen van een led	13
6.1	Challenges	13
6.1.1	Blink onboard LED challenge	13
6.1.2	Functie reference challenge	13
6.1.3	Blink external LED challenge.....	13
6.1.4	Fast blink LED challenge.....	13
6.1.5	Serial.println() challenge.....	13
6.1.6	Alternate LED blink challenge	13
7	Variabelen en printen van variabelen	14

7.1	Challenges	14
7.1.1	Variable challenge.....	14
7.1.2	Variables en datatypen	14
7.1.3	Variables en max/min waardes	14
7.1.4	Variables in het geheugen	14
8	Input – inlezen van een button en keuzestructuren	15
8.1	Challenges	15
8.1.1	Button challenge	15
8.1.2	Function reference challenge	15
8.1.3	If-else reference challenge.....	15
8.1.4	Button challenge	15
8.1.5	Button challenge vervolg	15
8.1.6	Two button challenge	15
8.1.7	Button expert challenge	16
9	Combinatie van input en output, flowcharts.....	17
9.1	Challenges	17
9.1.1	Three LED manual counting challenge	17
9.1.2	Three LED manual counting challenge with reverse challenge	17
10	Herhalingsstructuren	18
10.1	Challenges.....	18
10.1.1	For loop blink 3 times challenge	18
10.1.2	While loop blink 3 times challenge	18
10.1.3	LED fade expert challenge	18
11	Functies.....	19
11.1	Challenges.....	19
11.1.1	LED blink 3 and 5 times challenge	19
11.1.2	Function LED blink x times challenge.....	19
11.1.3	Button function expert challenge	19
12	Arrays	20
12.1	Challenges.....	20
12.1.1	Sum of integers array challenge	20
12.1.2	Record and play array challenge.....	20
12.1.3	Simon says expert challenge.....	20
12.1.4	Vault expert challenge	20
13	Analoge input en analoge output	22

13.1	Challenges.....	22
13.1.1	Analog input challenge	22
13.1.2	Living Color expert challenge.....	22
14	Communicatie tussen de Arduino en je laptop	23
14.1	Challenges.....	23
14.1.1	Calculator with Serial Monitor input challenge	23
14.1.2	Calculator with Serial Monitor input expert challenge	24
14.1.3	Vault with Serial Monitor input expert challenge	24
15	Eigen project	25

1 Inleiding

In dit document maak je kennis met de basis van C programmeren op een Arduino board. Ook leer je hoe je communicatie opzet tussen een Arduino en je laptop.

Dit doe je aan de hand van aangeleverde bronnen en challenges. Uitgewerkte challenges leveren zgn. proof of concepts op: stukken code / kleine programma's met een toelichting waarmee je een onderdeel van de leeruitkomst aantoonst. Het is de bedoeling dat je uiteindelijk met de proof of concepts en een eigen project de onderdelen van de leeruitkomst van deze cursus meerdere keren aantoonst.

1.1 Aangeleverde bronnen

De belangrijkste bronnen voor deze cursus staan genoemd in dit document. Deze bronnen bieden voldoende informatie om deze cursus succesvol af te ronden. Raadpleeg dus eerst deze bronnen voordat je verder gaat zoeken of vragen!

1.2 Demand based opzet

Aangezien deze cursus in demand based vorm is opgezet, kun je de volgende challenges gebruiken voor de proof of concepts:

- Basis challenges. Dit zijn de meeste challenges die in dit werkdocument worden gegeven. Als je deze maakt heb je voldoende basis materiaal om samen met je eigen project de leeruitkomst aan te tonen.
- Expert challenges. Een aantal challenges zijn gemarkeerd als 'Expert' challenge. Deze gaan wat verder als de leeruitkomst en zijn niet strikt noodzakelijk om de leeruitkomst aan te tonen. Maak deze alleen als je de basis challenges je gemakkelijk afgaat. Ook als je door wilt gaan met Technology zijn deze challenges aan te raden.
- Eigen challenges. In plaats van de gegeven challenges mag je ook je eigen challenges bedenken om de leeruitkomst aan te tonen. Overleg in dat geval wel met de docent om te checken of je challenges de leeruitkomst voldoende afdekken.

Let op: het is aan jouw de keuze of je de basis challenges wel of niet maakt, maar zorg ervoor dat je bij het overslaan van challenges geen leerdoelen overslaat en deze op een andere manier aantoonst.

- Je hoeft niet alle gemaakte challenges te documenteren of alle gemaakte code in te leveren. Lever alleen de challenges in waarvan je denkt dat die een zinnig proof of concept opleveren ofwel waarmee je duidelijk onderdelen van de leeruitkomst aantoonst.
- Standaard gebruiken we:
 - Arduino board en een Rich Shield.
 - Arduino C(++) programmeertaal.
 - Visual Studio Code met de PlatformIO extensie.

Je kunt je ook gebruikmaken van andere tools. Voorbeelden zijn:

- Scratch voor Arduino met mBlock i.p.v. Arduino C(++). Als je doorgaat met Software of Technology is het wel verstandig tijdig over te stappen naar een textuele programmeertaal.
- Arduino IDE i.p.v. Visual Studio Code. Deze is wat simpeler maar biedt minder mogelijkheden. Voor de oriënterende fase is dit echter prima.
- Breadboard met losse componenten. Hiermee heb je wat meer mogelijkheden, zeker als je een project gaat maken.
- Ander board als Arduino. Er zijn talloze alternatieven. Bedenk wel eerst waarom je een ander board wilt gebruiken.

Overleg hierover met je docent welke je daarbij kan begeleiden.

2 Installatie Arduino

Als eerste gaan we de Arduino programmeeromgeving installeren.

Als je een Arduino met een CH340 USB chip gebruikt, zoals uit de Technology Startsemester-Kit, moet je waarschijnlijk nog onder Windows de CH340 driver installeren, anders wordt je Arduino board niet herkend. De driver kun je vinden in de Canvas cursus onder Modules -> Oriëntatie toolbox -> CH341SER.ZIP.

Voor de programmeeromgeving kun je het volgende overwegen:

1. Standaard gaan we aan de slag met Visual Studio Code en de PlatformIO extensie. Visual Studio Code is een professionele IDE welke veel overeenkomsten vertoont met Visual Studio.
2. Als je nog weinig of geen programmeerervaring hebt kun je kiezen voor de standaard Arduino IDE. Deze is laagdrempelig maar heeft wel beperkingen in het gebruik.

Overleg met je docent als je niet zeker bent over je keuze.

2.1 Installatie Arduino IDE

Download en installeer de Arduino IDE van <https://www.arduino.cc/en/software>.

2.2 Installatie Visual Studio Code met PlatformIO

- Download en installeer Visual Studio Code en de PlatformIO extensie zoals beschreven op <https://docs.platformio.org/en/latest/ide/vscode.html>.
- Alleen voor als je libraries wilt toevoegen: libraries kunnen worden geïnstalleerd via het PlatformIO symbool  en dan QUICK ACCESS -> PIO Home -> Libraries. Typisch komen deze dan terecht in C:\Users\<username>\.platformio\lib. Het is ook mogelijk libraries hier handmatig naar toe te kopiëren. Na het opnieuw laden van het project worden deze libraries automatisch toegevoegd aan de .vscode\c_cpp_properties.json file zodat Visual Studio Code ze kan vinden. Let op: na het toevoegen van een nieuwe library kunnen er zogenaamde 'squiggles' (golflijntjes) bij de include regel komen met een '#include errors' melding. Als de library op de juiste manier is toegevoegd moeten deze squiggles verdwijnen na het opnieuw laden van het project.

2.3 Arduino Simulatie omgeving

Naast een fysieke Arduino is het ook handig om een simulatie omgeving te hebben. In deze cursus kun je gebruik maken van <https://www.tinkercad.com/> voor het uitproberen van kleine programma's. Een nadeel van deze simulator is wel dat je geen goede IDE tot je beschikking hebt. Gebruik het dus alleen voor kleine programma's. Voor de grotere programma's en het aantonen van de leeruikomst gebruik je een echte Arduino.

3 Hello World! op Arduino

Het eerste programma in een nieuwe programmeeromgeving is meestal een programma dat "Hello World!" uitprint. Dit is dan ook je eerste challenge!

Let op: in het onderstaande worden de stappen voor Visual Studio Code meer in detail genoemd dan voor de Arduino IDE omdat daar wat meer uitleg bij nodig is. Maak je gebruik van de Arduino IDE, kijk dan eerst op <https://docs.arduino.cc/software/ide-v1/tutorials/arduino-ide-v1-basics> om vertrouwd te raken met de omgeving. Er zijn ook vele YouTube tutorials zoals <https://www.youtube.com/watch?v=fJWR7dBuc18>. Vraag de docent als je problemen hebt met de Arduino IDE.

3.1 Challenges


3.1.1 Helloworld challenge

We beginnen met een simpel HelloWorld programma.

Maak je gebruik van de Arduino IDE en ben je er vertrouwd mee, dan kun je onderstaand programma overtypen en uitvoeren. De eerste regel '#include Arduino.h' is daarbij niet nodig. Dit programma zal 'Hello World' uitprinten over de USB kabel naar je laptop. Maak je nog even niet druk hoe dat precies gebeurt, dat komt later! De regels die met '//' beginnen zijn commentaar en doen verder niets. Deze mag je dus weglaten. In de Serial Monitor zal de tekst 'Hello World!' verschijnen.

```
1  #include <Arduino.h>
2
3  void setup() {
4      // put your setup code here, to run once:
5      Serial.begin(9600);
6      Serial.println("Hello World!");
7  }
8
9  void loop() {
10     // put your main code here, to run repeatedly:
11 }
```

Maak je gebruik van Visual Studio Code voer dan de volgende stappen uit:

- Ga naar  -> QUICK ACCESS -> PIO Home en klik op 'Open'. Maar daar een nieuw project aan genaamd 'HelloWorld'. Selecteer daarbij 'Arduino Uno' als board.
- Het nieuwe project bevat ook een 'platformio.ini' file. Hier staat in welke configuratie je gebruikt. Als het goed is zie je daar een configuratie als in het onderstaande screenshot: Onder Windows zou PlatformIO zelf de juiste poort moeten selecteren om naar je

```
[env:uno]
platform = atmelavr
board = uno
framework = arduino
```

Arduino te uploaden. Mocht dat om een of andere reden niet werken kun je zelf de

regel 'upload_port = COM3' toevoegen als COM3 je Arduino poort is. Op een Mac kun je 'upload_port = /dev/cu.wchusbserial*' toevoegen.

- Open de Explorer met het  symbool en ga naar 'HelloWorld -> src -> main.cpp'. Als het goed is opent main.cpp nu in de code editor. main.cpp bevat je programma dat we gaan aanpassen.
 - Typ het programma hierboven over in main.cpp.
 - Nu is het tijd om je eerste programma te bouwen en te uploaden naar de Arduino en te kijken of het werkt. Omdat het Arduino programma 'Hello World!' uitprint is het handig om een Serial Monitor te starten in de Visual Studio Code omgeving. Dit kun je doen met  -> PROJECT TASKS -> Monitor.
 - Je kunt je programma bouwen en uploaden met  -> PROJECT TASKS -> Upload. Als het goed is zie je daarna de 'Hello World!' tekst in het terminal window op je laptop.
- Opmerking:** De Serial Monitor blijft actief dus de volgende keer is  -> PROJECT TASKS -> Upload voldoende.

3.1.2 Serial.println challenge

Print eens een andere tekst uit en kijk wat er gebeurt als je de Serial.println in loop() zet.

4 Standaard programma structuur

In deze cursus maken we o.a. gebruik van de bronnen

<https://startingelectronics.org/software/arduino/learn-to-program-course/>

en

<https://www.arduino.cc/>.

Een duidelijke Nederlandstalige bron met basis uitleg is <https://arduino-lessen.nl/>.

In de onderstaande challenges gaan we hier kennis mee maken.

Opmerking: De meeste bronnen gaan uit van de standaard Arduino IDE. In deze cursus gebruiken we een meer professionele programmeeromgeving. De bediening is natuurlijk anders maar de voorbeelden werken verder hetzelfde.

4.1 Challenges

4.1.1 Program course challenge

Ga naar <https://startingelectronics.org/software/arduino/learn-to-program-course/> en bekijk de verschillende hoofdstukken (Parts) zodat je weet wat je er allemaal kunt vinden.

4.1.2 Structure of Arduino sketch challenge

Lees in Part 1 van <https://startingelectronics.org/software/arduino/learn-to-program-course/> hoe een standaard Arduino programma (sketch) is opgebouwd.

4.1.3 Function reference challenge

Zoek op <https://www.arduino.cc/reference/en/> de functies op die je hebt gebruikt in je HelloWorld programma en lees wat ze doen!

5 Hello World! op Arduino met Rich Shield

Nu wordt het tijd om het Rich Shield op de Arduino te monteren. Plaats het Rich Shield zodanig dat de blauwe DHT11 sensor aan de kant van de USB connector zit, zie foto hieronder.



5.1 Challenges

5.1.1 RichShieldHelloWorld challenge

Download de 'RichShieldHelloWorld' folder uit Modules -> Oriëntatie toolbox -> Example Programs als een .zip file naar je laptop en pak het uit. Laad het programma in Visual Studio Code. Dit kun je doen door de 'RichShieldHelloWorld' folder in Visual Studio Code te slepen. Let er hierbij wel op dat het padnaam naar die folder niet te lang is, anders kan Windows daar niet mee omgaan! Build en upload het programma naar de Arduino. Activeer ook de Serial Monitor. Als het goed is gaan er twee LED's branden en verschijnt er '1234' op het display en 'Hello World!' op de Serial Monitor. Bij het drukken op de knoppen gaan ook de andere twee LED's branden.

Note: mocht je het RichShieldHelloWorld voorbeeld in de Arduino IDE willen laden dan moet je:

- De main.cpp renamen naar main.ino.
- De src folder renamen naar main.
- Daarna kun je main.ino openen in de Arduino IDE.

Belangrijk: Het RichShieldHelloWorld programma is alleen om je Rich Shield uit te testen en als voorbeeld hoe je bijv. het display kunt aansturen. Ga bij de onderstaande challenges uit van een leeg project en dus niet van het 'RichShieldHelloWorld' programma. Dit om verwarring te voorkomen en zeker te weten dat het programma alleen jouw eigen code uitvoert.

Belangrijk!

De challenges die hieronder zijn beschreven gaan niet persé over een Arduino met een Rich Shield. Ze kunnen ook worden uitgevoerd met een Arduino met breadboard en losse componenten of zelfs met een ander board als Arduino. Als je het Rich Shield gebruikt geldt het volgende:

- *Als er wordt gesproken over bv. het aansluiten van een LED of button, hoef je dat bij het Rich Shield natuurlijk niet meer te doen.*
- *In de RichShield folder in de Oriëntatie toolbox staat een 'RichShieldPorts.pdf' met daarin de aansluitingen van het RichShield. Daar zie je bv. dat de rode LED op pin 4 zit aangesloten.*
- *Bij de buttons op het Rich Shield zit geen weerstand. Daarom moet je op de Arduino een zgn. pull-up weerstand activeren. Dit wordt later bij het gebruik van buttons verder toegelicht.*

6 Output – aansturen van een led

Je weet nu hoe je een Arduino programma kunt maken en uploaden. Het wordt nu tijd om de Arduino iets te laten aansturen, bv. een LED.

6.1 Challenges

6.1.1 Blink onboard LED challenge

Gebruik <https://docs.arduino.cc/built-in-examples/basics/Blink> als bron om de onboard LED aan te sluiten en te laten knipperen. Bij het gebruik van het Rich Shield is de onboard LED op het Arduino board lastig te zien.

6.1.2 Functie reference challenge

Zoek op <https://www.arduino.cc/reference/en/> de functies op die je hebt gebruikt in je programma en lees wat ze doen!

6.1.3 Blink external LED challenge

Sluit nu een externe LED aan en pas je programma aan zodat deze weer gaat knipperen. Kun je tegelijk de onboard LED ook nog laten knipperen? Als je een Rich Shield gebruikt hoef je zelf geen LED aan te sluiten maar gebruik je één van de LED's op het Rich Shield.

Tip (als je geen Rich shield gebruikt):

Om externe componenten aan te sluiten kun je een breadboard gebruiken. Op <https://www.youtube.com/watch?v=ufsMsXnG-y4> is te zien hoe een breadboard werkt.

6.1.4 Fast blink LED challenge

Kun je de LED sneller laten knipperen?

Bewijs dat de led sneller knippert door de tijd uit te printen (zie

<https://www.arduino.cc/reference/en/language/functions/time/millis/>)

6.1.5 Serial.println() challenge

Maak gebruik van de Serial.println() functie om “LED aan” en “LED uit” uit te printen.

6.1.6 Alternate LED blink challenge

Sluit twee LED's aan en laat ze om en om knipperen.

Tip:

Met een delay() functie kun je het programma laten wachten. Kijk op

<https://www.arduino.cc/reference/en/> voor een uitleg van de delay() functie.

7 Variabelen en printen van variabelen

Nu is het tijd om te oefenen met variabelen en het printen daarvan naar de Serial Monitor. Bekijk op <https://startingelectronics.org/software/arduino/learn-to-program-course/> de delen over variabelen en arithmetic operators. Probeer de voorbeelden zelf uit. Bekijk ook de volgende bron, <https://roboticsbackend.com/arduino-variable-types-complete-guide/>. Voor een video over het gebruik van de Serial Monitor, zie <https://www.youtube.com/watch?v=2JImBb9YLdg>. Daarna ben je klaar voor de volgende challenges!

7.1 Challenges

7.1.1 Variable challenge

Maak een programma dat een variabele 'count' maakt en deze elke seconde met één ophoogt en vervolgens print naar de Serial Monitor. Stel dat deze variabele count het aantal mensen op een festivalterrein weergeeft, welk datatype zou deze variabele dan kunnen hebben en waarom? Probeer het uit in je programma.

7.1.2 Variables en datatypen

Zoek uit welke datatypen er allemaal zijn binnen de arduino. Maak variabelen aan met deze datatypen in een programma en print deze uit naar de seriële monitor. Verzin voor elk datatype een situatie waarin je deze variabele zou kunnen gebruiken. Waarvoor zijn constanten bedoeld? Wat betekent signed en unsigned? Laat zien met een voorbeeld.

7.1.3 Variables en max/min waarden

Zoek uit voor elk datatype wat de maximale waarde en de minimale waarde is die je erin kunt opslaan en print deze waarde uit in de seriële monitor. Wat gebeurt er als de waarde die je in de variabele wil opslaan te groot is? Probeer dit uit en laat dit zien. Uit hoeveel bits bestaat elk datatype? Wat heeft het aantal bits te maken met de maximale waarde die je erin kunt opslaan?

7.1.4 Variables in het geheugen

Gebruik variabelen van verschillende datatypes om waarden in op te slaan en laat met `Serial.println(<variabele>, BIN)` zien hoe deze waarde wordt opgeslagen in het geheugen. Dit heet de binaire representatie van een getal.

8 Input – inlezen van een button en keuzestructuren

Voor de meeste projecten is het nodig om de Arduino te kunnen bedienen bv. d.m.v. een button. Daarbij komt automatisch het 'if' statement om de hoek kijken. Je wilt namelijk dat je programma iets doet **als** (if) de button wordt ingedrukt. Om te leren hoe je een button aansluit en keuzes kunt maken in je programma met een 'if' en 'else' statement kun je de onderstaande challenges maken.

8.1 Challenges

8.1.1 Button challenge

Gebruik <https://docs.arduino.cc/built-in-examples/digital/Button> om een **externe** LED te bedienen met een button. Als je een Rich Shield gebruikt gebruik je een LED en een button op het Rich Shield. Je moet daarbij wel een zgn. pull-up weerstand op de Arduino activeren. Dit kun je doen door i.p.v. `pinMode(buttonPin, INPUT), pinMode(buttonPin, INPUT_PULLUP)` te gebruiken.

8.1.2 Function reference challenge

Zoek op <https://www.arduino.cc/reference/en/> de functies op die je hebt gebruikt in je programma en beschrijf wat ze doen!

8.1.3 If-else reference challenge

Bekijk <https://startingelectronics.org/software/arduino/learn-to-program-course/> en <https://www.arduino.cc/reference/en/#structure> voor een uitleg over het if en else statement.

8.1.4 Button challenge

Een festivalterrein heeft 1 ingang. De organisatie wil bijhouden hoeveel mensen er op het terrein aanwezig zijn. Hiervoor wordt er elke keer als er iemand het festivalterrein opkomt een knop ingedrukt.

Maak een systeem met de arduino waarmee je mensen kunt tellen. Gebruik hiervoor een button en lees de pin waarmee de button verbonden is uit. Als de button is ingedrukt dan hoog je een variabele op. Zorg er ook voor dat je in de seriële monitor laat zien hoeveel mensen er op het terrein zijn en op welk moment ze zijn binnengekomen

Wat valt je hierbij op?

8.1.5 Button challenge vervolg

Vaak wil je dat bij '1 keer indrukken van een button maar 1 keer een actie wordt uitgevoerd. Een pragmatische oplossing is om een kleine delay toe te voegen nadat een button press is gedetecteerd. Bij het goed kiezen van de delay is de button door de gebruiker alweer losgelaten als het programma de volgende loop uitvoert. Pas een `delay()` functie toe in je programma en kies de delay zodanig dat bij het kort indrukken van de button de variabele maar met 1 wordt opgehoogd wordt uitgeprint en het programma toch snel genoeg blijft reageren.

Test je programma. Wat gebeurt er als je de delay te lang of te kort maakt? Probeer dat uit
Wat is het maximale aantal mensen dat je kunt tellen op het festivalterrein met het datatype wat je hebt gekozen?

Kun je het datatype zo aanpassen dat je er meer in kunt opslaan? Laat zien

Wat gebeurt er als de maximale waarde van de variabele bereikt wordt? Probeer dat uit

8.1.6 Two button challenge

Kun je het voorbeeld uitbreiden naar twee buttons?

Als de mensen van het festivalterrein af willen moeten ze naar de uitgang lopen. Uiteraard moet dan het aantal mensen op het terrein worden aangepast via hetzelfde systeem, met een tweede button. Pas je systeem zo aan dat als een festivalganger het terrein verlaat, het aantal bezoekers met een druk op de tweede knop wordt aangepast.

Hoe voorkom je dat het aantal festivalgangers negatief wordt in je programma? Vind hier een oplossing voor en test deze uit

8.1.7 Button expert challenge

Het uitlezen van buttons met een delay zoals hierboven beschreven is niet optimaal. Je programma wordt vertraagd door de delay en je mag de button niet te lang indrukken. Voor de oriënterende fase is deze oplossing echter ok. Als je als expert met buttons om wilt gaan kun je het voorbeeld op <https://docs.arduino.cc/built-in-examples/digital/StateChangeDetection> bekijken.

En als je een echte professional bent, vervang je de debounce delay in dit voorbeeld door het gebruik van `millis()`.

Pas het vorige programma aan met `statechangedetection`, zorg ervoor dat je de code in het voorbeeld ook echt goed begrijpt

9 Combinatie van input en output, flowcharts

Op dit moment kun je omgaan met een Arduino, LED's, buttons, variabelen en je kunt iets printen naar de Serial Monitor. Gebruik de volgende challenges om je vaardigheid te testen en te vergroten. Je mag ook een eigen variatie op onderstaande challenges maken. Gebruik echter nog geen eigen while of for loops (dit komt later), alleen de standaard 'void loop()' functie. Zoek specifieke commando's op in <https://www.arduino.cc/reference/en/>.

Omdat de programma's die je maakt steeds ingewikkelder worden ga je vanaf hier ook werken met flowcharts, zie de presentatie in de 'Toolbox' sectie onder 'Workshops'. Dit hoeft niet bij elke challenge, maar maak een flowchart als de structuur van je programma niet meteen duidelijk is, zie ook <https://www.arduino.cc/reference/en/>.

9.1 Challenges

9.1.1 Three LED manual counting challenge

De organisatie van het festival van de vorige challenge wil graag een uitbreiding op het systeem met LEDs. Als er een festivalganger het terrein opkomt dan moet bij een druk op de knop een volgende LED gaan branden. Na LED nummer drie licht LED nummer één weer op. De volgorde is dus 1 – 2 – 3 – 1 – 2 enz.

LET OP! Deze challenge is misschien lastiger dan je denkt. Probeer er eerst zelf uit te komen en vraag de docent om advies indien nodig!

9.1.2 Three LED manual counting challenge with reverse challenge

De festivalorganisatie zou graag willen dat als een festivalganger het festival verlaat niet alleen het aantal mensen op het festival wordt aangepast met een druk op de knop, maar ook dat de richting van het oplichten van de LED's omkeert, de volgorde wordt dus 3 – 2 – 1 – 3 – 2 enz. Dit is een uitbreiding van challenge 9.1.2, dus de vorige functionaliteit moet ook nog gewoon werken.

10 Herhalingsstructuren

Nu is het tijd om te kijken naar herhalingsstructuren zoals de for en while loop. Bekijk deze in de cursus <https://startingelectronics.org/software/arduino/learn-to-program-course/>.

Oefen zelf met de gegeven voorbeelden!

10.1 Challenges

Bij deze challenges maak je gebruik van een for of while loop, samen met een if statement indien nodig. Programmeer de functionaliteit in de loop() functie, dus niet in de setup(). Maak ook flowcharts.

10.1.1 For loop blink 3 times challenge

De organisatie van het festival wil weer een uitbreiding op het systeem. Als het maximum aantal mensen bereikt is, dan moet er een rood lampje 3 keer knipperen om dit aan te geven. Doe dit met een eigen for loop.

Tip: gebruik een extra variabele waarin je het maximaal aantal mensen opslaat. Zorg ervoor dat je dit goed test

10.1.2 While loop blink 3 times challenge

Doe hetzelfde nu met een while loop.

10.1.3 LED fade expert challenge

Sluit een LED aan en laat de LED langzaam van uit naar vol aan faden en dan weer naar uit enz. Gebruik hierbij 2x een for loop. Om te faden heb je de analogWrite() functie nodig.

Hierover later nog meer. Gebruik

<https://www.arduino.cc/en/Tutorial/BuiltInExamples/Fade> voor uitleg en een voorbeeld.

Let op: dit voorbeeld is zonder for loops, pas het programma dus hierop aan! Als dat werkt kun je je creativiteit gebruiken voor een mooie lichtshow voor op het festival!

11 Functies

Een functie in de Arduino programmeertaal (C/C++) is te vergelijken met een methode in C#. Als je hier bent aangekomen heb je al aardig wat functies gebruikt, bv. `digitalRead()`, `digitalWrite()`, `delay()`, `Serial.println()` enz. Zie <https://www.arduino.cc/reference/en/> voor een overzicht van alle Arduino functies. Voor een uitleg van functies en hoe je ze zelf kunt maken, zie <https://startingelectronics.org/software/arduino/learn-to-program-course/>.

Voor een video over functies, zie

https://www.youtube.com/watch?v=tft5RnrG4JU&ab_channel=BartBozon.

11.1 Challenges

Bij de onderstaande challenges ga je zelf met functies aan de slag. Als je het goed aanpakt kun je een aantal van deze functies de rest van je Arduino carrière blijven gebruiken!

11.1.1 LED blink 3 and 5 times challenge

Sluit één led aan en twee buttons. Maak een ledknipper functie om een LED 3x te laten knipperen als button één wordt ingedrukt.

11.1.2 Function LED blink x times challenge

Nu wordt er gevraagd om de led 5x te laten knipperen als button twee wordt ingedrukt. Bedenk zelf hoe je dit het handigst kan oplossen.

Hint: gebruik een parameter!

11.1.3 Button function expert challenge

Deze challenge bouwt voort op Expert challenge 0.

Omdat bijna elk Arduino project gebruik maakt van buttons is het handig om voor het uitlezen van een button een functie te maken. Het is dan ook de moeite waard om wat tijd de steken in het maken van een goede functie. Maak een `buttonread` functie die voldoet aan de volgende eisen:

- Je programma reageert alleen op het indrukken van de button, dus niet op het loslaten.
- Het maakt voor je programma niet uit of je de button kort of lang ingedrukt houdt.
- Er wordt geen `delay()` functie gebruikt.
- Je mag per button aan aparte functie maken met eigen variabelen. Later zul je zien dat dit overeenkomt met het maken van een button class in C#.

12 Arrays

Om een 'rijtje' van variabelen op te slaan kun je een array gebruiken. Voor een uitleg van arrays en hoe je ze zelf kunt maken, zie

<https://startingelectronics.org/software/arduino/learn-to-program-course/>. Hoe je arrays gebruikt om een aantal LED's aan te sturen zie je op <https://docs.arduino.cc/built-in-examples/control-structures/Arrays>.

12.1 Challenges

12.1.1 Sum of integers array challenge

Maak een programma waarbij je 5 integers in een array opslaat en daarna de som berekend van de 5 opgeslagen waarden. Vervolgens worden de 5 waarden uitgeprint en ook de som.

12.1.2 Record and play array challenge

Sluit twee buttons aan (button 1 en 2) en 2 LED's (LED 1 en 2). Schrijf een programma dat 4 button presses opslaat in een array en deze daarna op de LED's afspeelt. Voorbeeld: Als de gebruiker button sequence 1-2-2-1 indrukt gaan daarna de LED's branden in de volgorde 1-2-2-1.

Tip:

Gebruik `Serial.println()` om te checken of de button presses correct worden opgeslagen in de array.

Zelfcheck:

Zou je je programma eenvoudig kunnen uitbreiden naar bv. 8 buttons en 8 LED's? Zoja, dan snap je nu hoe arrays werken!

12.1.3 Simon says expert challenge

Breid bovenstaande challenge uit tot een Simon Says spel! Dit kun je doen door een buzzer toe te voegen die verschillende tonen afspeelt bij het indrukken van de buttons. Je kunt daarvoor de `tone()` functie gebruiken. Een tweede speler moet dan de melodie van de eerste speler naspelen.

12.1.4 Vault expert challenge

In deze challenge ga je een kluis maken welke je kunt openen met een pincode. Een array kun je natuurlijk mooi gebruiken om een pincode in op te slaan. Sluit twee buttons aan en een rode LED en een groene LED.

Schrijf een programma voor een kluis met de volgende eisen:



- Bij het opstarten van het programma gaat de rode LED branden en verschijnt de tekst 'voer pincode in' op de Serial Monitor.
- Vervolgens voert de gebruiker een 4-cijferige pincode in. Bedenk zelf een handige manier met twee buttons.
- Elke keer dat een cijfer is ingevoerd verschijnt het ook op de Serial Monitor.
- Nadat het vierde nummer is ingevoerd verschijnt de tekst 'pincode ingevoerd, checking...' op de Serial Monitor.

- Als de ingevoerde pincode overeenkomt met een vooraf opgeslagen pincode gaat de groene led voor één seconde branden en verschijnt de tekst 'kluis open'.
- Bij een foute pincode gaat de rode led 3 keer knipperen en verschijnt de tekst 'code onjuist'.
- Daarna gaat de rode led weer branden en verschijnt de tekst 'voer pincode in'.

Tips:

- Sla de geheime pincode op in een array en de ingevoerde pincode in een andere array.
- Voor het vergelijken van twee arrays kun je een for loop gebruiken.
- Maak voor het lezen van de buttons en voor het knipperen van de LED's gebruik van de functies die je al eerder hebt gemaakt en getest.

13 Analoge input en analoge output

Naast de digitale input en output (aan of uit) is het mogelijk om een continue waarde als input of output te hebben. Dit heet dan analoge input en output. Een voorbeeld is een potentiometer (draaiknop) waarmee je de helderheid van een LED kan instellen. Voor analoge input kun je <https://www.arduino.cc/en/Tutorial/BuiltInExamples/AnalogInput> bekijken en voor analoge output

<https://www.arduino.cc/reference/en/language/functions/analog-io/analogwrite/>. Voor de experts: een analoge output wordt gemaakt m.b.v. PWM (Pulse Width Modulation ofwel pulsbreedtemodulatie). Hoe analoge input en output precies werken hoeft je niet te weten, maar wel wat je ermee kunt doen. Je hebt dan ook wat meer mogelijkheden bij je eigen project!

13.1 Challenges

13.1.1 Analog input challenge

Sluit een potentiometer aan en een LED. Schrijf een programma waarmee je de LED van volledig aan naar volledig uit kunt dimmen door aan de potentiometer te draaien. Print ook de waarden van de potentiometer uit.

Tips:

- Analoge input en output werkt niet op alle pinnen van de Arduino. Gebruik voor analoge input de pinnen A0 t/m A5 en voor analoge output de pinnen met een ~ symbool ofwel pin 3, 5, 6, 9, 10 of 11 op een Arduino Uno.
- Je zult de waarde die uit de potentiometer komt (0..1023) moeten omzetten naar een waarde die nodig is om de LED in te stellen (0..255). Je kunt dit oplossen met een deling maar je kunt ook gebruik maken van de map functie, zie <https://www.arduino.cc/reference/en/language/functions/math/map/>.

13.1.2 Living Color expert challenge

Sluit nu een RGB LED aan en zorg ervoor dat je de kleur kunt instellen met de potentiometer. Onderzoek zelf hoe je zorgt voor een mooie kleurovergang.

14 Communicatie tussen de Arduino en je laptop

Je Arduino met programma kun je zien als een klein embedded systeem. Meestal moet zo'n systeem communiceren met andere systemen, bv. een server welke dan sensordata kan verzamelen. Je laat nu de Arduino communiceren met de Serial Monitor op je laptop.

- Open een nieuwe sketch in de Arduino IDE of creëer een nieuw project in Visual Studio Code.
- Download het voorbeeldprogramma 'ArduinoRemoteControlExample.cpp' uit de 'Toolbox' sectie en kopieer de inhoud naar je Arduino IDE sketch of de main.cpp file van je Visual Studio Code project.
- Bekijk het programma en lees het commentaar zorgvuldig.
- Het is belangrijk dat de instellingen van de Serial Monitor overeenkomen met de instellingen op de Arduino. Deze zijn in dit voorbeeld:
 - Een communicatiesnelheid (baudrate) van 9600 baud.
 - Na een enter wordt een newline of linefeed character ('\n') gestuurd richting Arduino.
 - Lokale echo aan zodat je kunt zien wat je intypt.

In de Arduino IDE kun je dit in het Serial Monitor window instellen. De lokale echo staat daar al aan.

In Visual Studio Code met PlatformIO kun je dit regelen door het toevoegen van de monitor_echo setting in de platformio.ini file als in onderstaand screenshot.

```
[env:uno]
platform = atmelavr
board = uno
framework = arduino
monitor_speed = 9600
monitor_echo = yes
```

- Maak je gebruik van de Arduino IDE, typ dan in de Serial Monitor Send textbox 'LED on' in en klik op 'Send'.
- Maak je gebruik van Visual Studio Code, typ dan in de Serial Monitor window 'LED on' en daarna enter.
- Nu wordt de message 'LED on' naar de Arduino verzonden. Deze zet in de voorbeeld Arduino code de onboard LED aan en deze stuurt ter bevestiging weer 'LED on' terug'. Dit is te zien in de Serial Monitor, zie als voorbeeld het onderstaande Visual Studio Code screenshot.

```
--- Miniterm on COM3 9600,8,N,1 ---
--- Quit: Ctrl+C | Menu: Ctrl+T | Help: Ctrl+T followed by Ctrl+H ---
LED on
Arduino Received: LED on
LED off
Arduino Received: LED off
```

- Nu weet je dat de communicatie werkt en kun je zelf in de Arduino code de communicatie aanpassen naar de messages die je wilt gebruiken. Zoek hiervoor naar "TODO" in 'ArduinoRemoteControlExample.cpp'.

14.1 Challenges

14.1.1 Calculator with Serial Monitor input challenge

Maak van de Arduino een eenvoudige calculator welke kwadraten kan uitrekenen. De gebruiker voert in de Serial Monitor een getal (integer) in gevolgd door een enter ('\n'). De Arduino berekent het kwadraat van het getal en stuurt het getal terug naar de Serial

Monitor.

Tip:

- De getallen die je invoert via de Serial Monitor komen in het Arduino voorbeeldprogramma terecht in een string. Met behulp van de `toInt()` kun je hier weer een getal van maken, zie <https://www.arduino.cc/reference/en/language/variables/data-types/stringobject/> voor deze en andere handige string functies.

14.1.2 Calculator with Serial Monitor input expert challenge

De bovenstaande calculator is natuurlijk wel erg simpel. Als expert ga je hem uitbreiden zodat hij ook kan optellen en vermenigvuldigen. Daarbij kunnen eenvoudige sommen worden uitgerekend als '42 + 12' of '42 * 12'. Het antwoord wordt dan door de Arduino teruggestuurd naar de Serial Monitor.

Tip:

- Maak het jezelf niet te moeilijk. Laat de gebruiker de som invoeren als twee getallen en daarna de gewenste operatie met telkens een enter ('\n') ertussen. Voor het '42 + 12' voorbeeld: '42 <enter> 12 <enter> + <enter>'. Dit is de zogenaamde Reverse Polish Notation welke op sommige rekenmachines wordt toegepast zodat je geen haakjes nodig hebt. Op deze manier kun je eerst de getallen opslaan en daarna de gewenste operatie uitvoeren.

14.1.3 Vault with Serial Monitor input expert challenge

In deze challenge ga je de kluis challenge uitbreiden zodanig dat deze ook vanaf de Serial Monitor bediend kan worden. Dit betekent dat je voor het invoeren van de pincode zowel de buttons als de Serial Monitor kunt gebruiken.

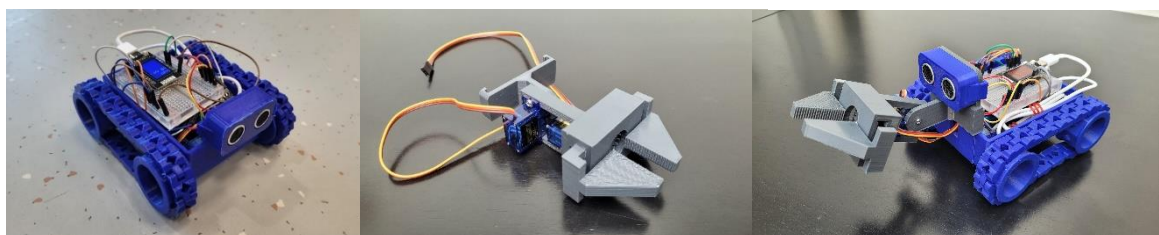
15 Eigen project

Als je hier bent aangekomen, weet je hoe je een C programma kunt maken en uitvoeren op een Arduino. Ook weet je hoe je een Arduino met een applicatie op je laptop kunt laten communiceren. Dit kan een Serial Monitor zijn of een eigen C# applicatie. Tijd om een project te bedenken dat gebruik maakt van sensoren en actuatoren en waarmee je het geleerde nog eens kunt demonstreren!

Dit is ook een mooie gelegenheid om eens te werken met andere actuatoren en sensoren dan LED's en buttons. Je kunt gebruiken wat er op het RICH Shield zit of je gaat aan de slag met een breadboard en losse componenten. Een selectie van wat er beschikbaar is bij de ISSD vind je in Modules -> Oriëntatie toolbox -> HardwareList.

Schrijf eerst zo concreet mogelijk op wat je project inhoudt en vraag feedback aan de docent. Om je op weg te helpen staan hieronder wat voorbeeld projecten.

- Remsysteem: zodra een object te dichtbij komt (afstandssensor) wordt er op de rem getrapt (buzzer of servo). De toegestane afstand is in te stellen vanaf de laptop applicatie en de actuele afstand is daar ook zichtbaar.
- Temperatuurregeling: zodra de temperatuur boven een bepaalde waarde komt (temperatuursensor) worden de ramen opengezet (display of servo). De toegestane temperatuur is in te stellen via de laptop applicatie en de actuele temperatuur is daar ook zichtbaar.
- Lichtshow: Een aantal LEDs geven verschillende lichtpatronen weer. Het lichtpatroon is in te stellen vanaf de laptop applicatie. De Arduino meldt terug aan de laptop applicatie welk lichtpatroon er actief is.
- Verkeerslichtregeling voor een kruispunt. Met buttons wordt verkeer gesimuleerd. De laptop applicatie geeft aan welke rijbaan groen heeft. Vanaf de laptop applicatie kunnen alle verkeerslichten op een **veilige manier** op knipperen worden gezet.
- Project met MiniBot en / of RobotGripper



Als je wilt kun je je eigen mini robot en / of gripper uitprinten.

Onder 'Modules -> Oriëntatie -> 3D prints' vind je de benodigde print files en een '3D print guideline.pdf'. Daarin staat ook beschreven hoe je het ontwerp kunt aanpassen / uitbreiden naar je eigen wensen.

De RobotGripper kun je los gebruiken of op de MiniBot schuiven. De MiniBot kun je uitrusten met twee continuous rotation TS90D mini servo's en een ultrasonic sensor HC-SR04.

De RobotGripper werkt met twee TS90M mini servo's (metal gearing en geen continuous rotation). Zie onderstaande plaatjes.



Daarnaast kun je op een breadboard extra sensoren toevoegen.

Je kunt de MiniBot uitrusten met een Arduino Uno maar ook met bv. een ESP32 board.

Maak bv. een programma dat de MiniBot laat rijden zonder ergens tegenaan te rijden.

Als je een embedded board met Bluetooth kiest, kun je het programma zo maken dat je de MiniBot en / of RobotGripper kunt besturen met een app zoals

<https://play.google.com/store/apps/details?id=com.appsvalley.bluetooth.arduinocontroller>.

Voer het project uit en evalueer in hoeverre je project voldoet aan de verwachtingen!

SUCCESSION!