

Microcontroller

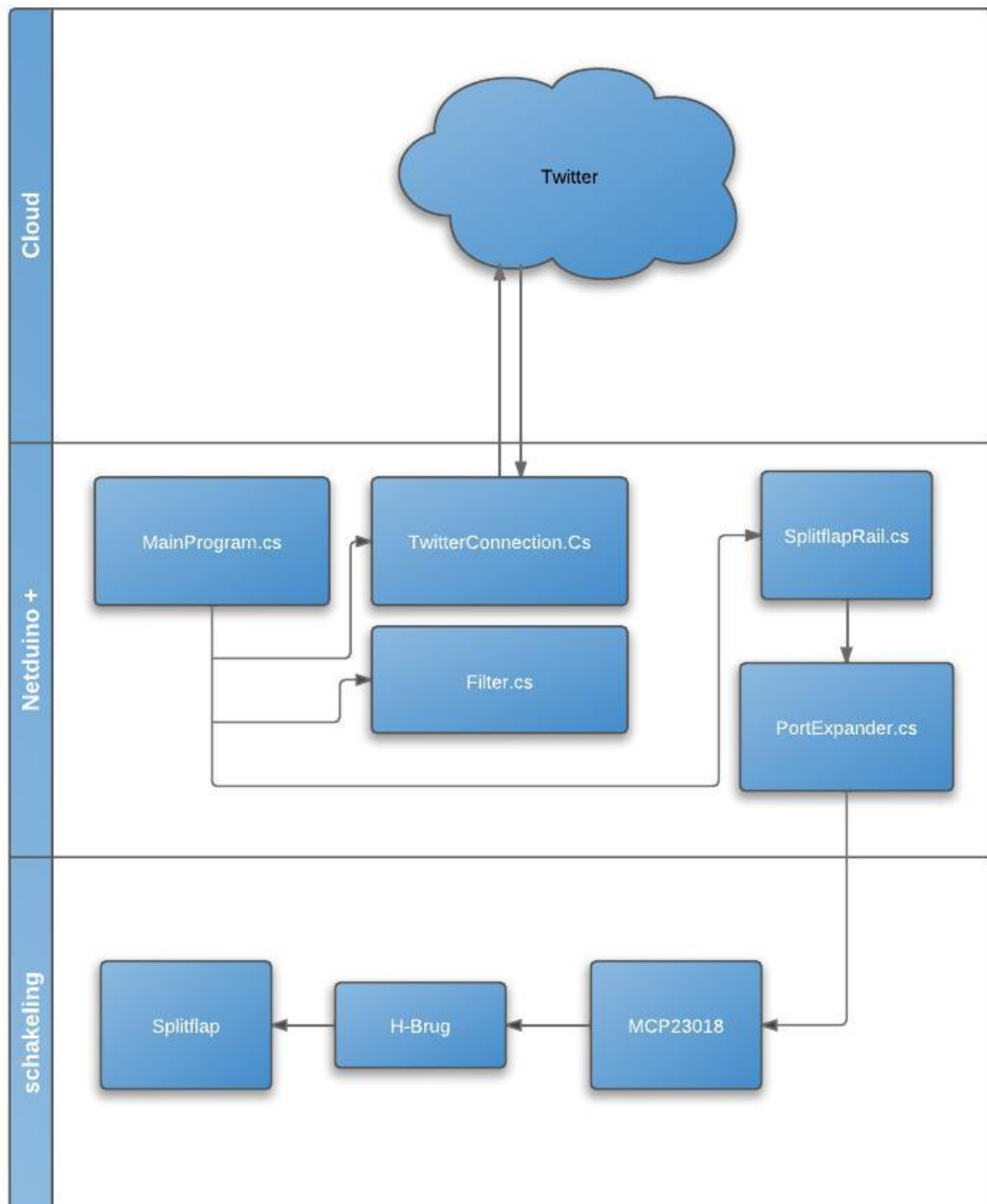
Netduino+

Netduino+ is een arduino microcontroller met het .net micro framework er op. Dit staat toe om je programma's op het .net framework te laten draaien. Je kan daardoor programmeren in visual C# en alle andere talen ondersteund door het .net micro framework.

.NET micro framework

Dit is een lichtere, kleinere versie van het .net framework. Doordat het zo licht is word bepaalde .net functionaliteit, zoals generische lijsten, niet ondersteund.

Programma



Afhalen van Twitter

Dit deel wordt afgeschermd in de klasse `TwitterConnection`, allereerst wordt de klasse aangemaakt met een hashtag, deze wordt aan de zoekquery toegevoegd

(vb. <http://search.twitter.com/search.json?q=%23ikdoeict>)

Er zijn verschillende methoden gemaakt voor de zoekquery aan te passen zoals `rpp` (results per page) en `since_id`.

Deze methoden roepen een `HttpWebRequest` op met de URL, de `HttpWebRequest` zorgt ervoor dat we de query uitvoeren op de twitter server, en we nemen een antwoord op met `WebResponse`.

Het resultaat van die `WebResponse`, een json, wordt geparsed door een nieuwe methode, het parsen gebeurt door middel van `IndexOf` en `Substring`.

Alle resultaten van de query (of als results per page = 1 (&rpp=1)) worden in een `ArrayList` geplaatst en doorgegeven aan de bovenliggende methoden, en die geeft ze dan terug aan de main loop.

Filter

Deze klasse leest `filter.txt` in dat op een SD kaart staat. In dit tekst bestand staan alle woorden die verboden zijn in een tweet. Eerst wordt de tekst uit een tweet object gehaald. Vervolgens worden de woorden gesplit en vergeleken met de slechte. Wanneer er een overeenkomst is wordt de tweet niet gebruikt.

Daarna wordt ook op de leestekens gefilterd omdat de splitflaps maar beperkte tekens bevatten. Wanneer er een teken voorkomt wordt deze vervangen door een spatie (lege flap). Vervolgens wordt de tweet tekst opnieuw opgebouwd. Uiteindelijk wordt er een tweet object gereturned.

Aansturing schakeling

I2CPlug

Deze klasse voorziet alle functionaliteit van een I2C device. Lezen en schrijven + lezen en schrijven van een register. Deze klasse is gebruikt de repeated-start-bit functionaliteit en is dus niet geschikt voor I2C apparaten die dat niet ondersteunen. De clock-rate wordt automatisch gezet op 400Kh.

PortexpanderPlug

Deze klasse erft over van de `I2CPlug` klasse en voorziet alle functionaliteit om de MCP23018 aan te sturen. In de constructor worden de pinnen automatisch op output gezet. Input is niet mogelijk.

Je geeft het adres van de chip mee met de constructor:

```
PortExpanderPlug(0x20);
```

Je schrijft een byte naar de pinnen (bit 0 = pin 0). Je geeft de kant van de MCP (A of B)

```
plug.WriteToPins(byte, Side);
```

Interne pullup wordt NIET aangezet, je moet die dus zelf nog aanzetten.

SplitflapConfig

Deze klasse houdt data van de splitflap bij. De beschikbare karakters en hun plaats worden statisch bijgehouden in de klasse. Verder wordt de plaats van de splitflap in de schakeling bijgehouden ahv portexpander adres, zijde van de portexpander en bit van de portexpander.

Het huidig getoonde karakter van de splitflap wordt ook bijgehouden. Bij het aanpassen van deze waarde wordt automatisch gekeken of de splitflap een volledige toer heeft gemaakt. De waarde wordt dan op 0 gezet.

SplitflapRail

Maakt een array van SplitflapConfig-objecten.

De splitflap wordt 1 stap verzet door de betreffende bit van de portexpander te toggelen. Momenteel wordt iedere splitflap apart verzet. Na iedere bit te verzetten moet er een periode gewacht worden tot de flap mechanisch op de volgende positie staat. Dit stond eerst op 50ms maar we hebben de wachttijd moeten vergroten naar 100ms omdat de flaps die er het slechtst aan toe waren anders niet mee konden.

De code kan geoptimaliseerd worden alle splitflaps na elkaar een stap te laten verzetten en pas daarna te wachten. De wachttijd zal dan gelijk zijn aan "normale wachttijd"/"aantal splitflaps"

je maakt een splitflaprail object en geeft de initiele staat mee aan de constructor

```
SplitflapRail(initialState);
```

Je geeft de splitflaprail een nieuwe text tonen

```
rail.show(nieuweTekst);
```

Sturing

MCP23018 port expander

We hebben voor de MCP23018 gekozen omdat deze 16 poorten heeft en via het I2C protocol werkt.

De MCP heeft een configureerbaar adres. De eerste 4 bits van het adres zijn statisch. De laatste 3 bits van het adres worden bepaald door de spanning aan de adres-pin.

I2C protocol

We hebben gekozen voor een chip die werkt met het I2C protocol ipv het SPI protocol. Het SPI protocol is niet bedoeld om een groot aantal slaves te hebben terwijl het I2C daar speciaal voor gemaakt is.

Schakeling

Het I2C werkt dmv open drain collectors. Ipv een positieve spanning te geven bij logische 1 drainen ze de aanwezige positieve spanning bij een logische nul. Dit om verschillen in de klok van verschillende masters op te lossen. Dit zorgt er wel voor dat alle ingangen en uitgangen een pull-up weerstan nodig hebben. Deze zet de spanning op het logisch-1 voltage als ze niet wordt gedraint

De MCP23018 heeft een inwendige pull-up weerstand aan zijn uitgangen. Deze kan aangezet worden door register GPPU op 255 te zetten. Iedere bit van GPPU zet de desbetreffende GPIO-pull-up resistor aan. We dit echter niet gebruiken. Het I2C protocol van netduino werkt op 3.3V. De MCP moet dus ook op 3.3V gevoed worden. De H-bruggen werken echter op 5V. We zetten dus een pull-up resistor van 5V aan de poorten zodat de MCP, gevoed op 3.3V, een output heeft van 0V-5V.

Digitale aansturing

De MCP werkt dmv de repeated start bit. Eerst wordt de startbit doorgestuurd gevolgd door het adres van het register dat je wil aanspreken. De startbit wordt daarna nog eens herhaald gevolgd door een byte met de waarde die het register moet krijgen.

De registers kunnen op 2 manieren aangesproken worden: byte mode en sequential mode. In byte mode zijn de registers 0x00 tot 0x07 voor configuratie van kant A en 0x08 tot 0x15 voor configuratie van kant B. Op deze manier is de MCP compatibel met code geschreven voor een 8-bit port expander. Wij gebruiken echter sequential mode. Daarbij zijn alle even registers voor configuratie van kant A en alle oneven registers voor configuratie van kant B. voor deze stand zet je IOCON op 0.

Via het GPPU kunnen we de interne pull-ups aanzetten, maar die functionaliteit gebruiken we niet.

IODIR zet je op 0x00 om de pinnen als output in te stellen.

Splitflap

De splitflap bestaat uit een stappenmotor en een sensor

Stappenmotor

De motor in de splitflap is een stappenmotor. Iedere keer de spanning omgewisseld word verspringt hij 1 flap. De frequentie waarmee de spanning omgewisseld word moet hoog genoeg zijn om de motor toe te laten de volledige stap te verspringen. Doordat het materiaal niet meer in topconditie is moet er tot 100ms gewacht worden voor de spanning mag omgeschakeld worden. Dit zorgt ervoor dat de flaps zeer traag verspringen.

Sensor

De sensor geeft aan wanneer de flap op 0 staat. Het grootste deel van de sensors werkt echter niet meer waardoor we die niet kunnen gebruiken.

H-brug

De spanning over de splitflap wordt geregeld door de L293D H-brug

Schakeling

De H-brug wordt gevoed op 5V. input 1, 2, 3 en 4 werken ook op 5V. output 1, 2, 3 en 4 geven zelfde spanning als VS bij logische 1.

