Besturingen vergelijking

# Inhoud

# Inleiding

In dit document ga ik over de besturing van de wasmachine project. We hebben nu de PLC gekozen echter ga ik dit nu vergelijken met 3 andere besturingen systeems.  
in dit geval heb ik gekozen voor een microcontroller (in dit geval vergelijk ik het aan de Arduino mega, een Rasberry pi 3 en een zelf ontworpen controller.   
  
als eerst ga ik kijken naar de benodigdheden voor de componenten. Een voorbeeld is dat de PLC 24 volt uitgangen heeft en een Arduino mega heeft alleen maat 5V uitgangen enz.   
sommige componenten moeten daarom ook via een extra connectie/besturing paneel nodig. Dit kost meer geld en gebruikt meer watt. Ook maakt het de machine minder robuust.  
  
de componenten waar ik naar ga kijken zijn de volgende.

|  |  |  |
| --- | --- | --- |
| **OUTPUT** | DATASHEET | aansluiting |
| Trommel motor | https://www.transmotec.com/Download/Datasheets/Transmotec-Datasheet-AI-025W-230-SCB.pdf | 230V |
| Relais warmte element | [FOTEK CONTROLS CO., LTD | SSR-40DA - Datasheet PDF & Tech Specs - Datasheets.com](https://www.datasheets.com/part-details/ssr-40da-fotek-controls-co---ltd-77834039#datasheet) | Min 3v to ground |
| Magneet ventiel | [Magneetventiel wasmachine vervangen | Zie hoe gemakkelijk het is »](https://nettoparts.nl/shop/diy-video-wasmachine-magneetventiel-8908s1.html?CookieConsentChanged=1) | 12v Neutraal  / i2c |
| deurslot | (geen datasheet) volgens het internet zitten er 4 draden op. Is de data draad +24V en ground. Slot gaat dicht wanneer bekrachtig | +24v ground |
|  |  |  |
| **INPUT** | DATASHEET |  |
| Waterniveausensor | [DSC-6B Elektronische Waterniveau Sensor Drukschakelaar Voor Samsung Voor Lg Trommel Wasmachine Rerpair Accessoires Onderdelen Dc 5V - AliExpress 6](https://nl.aliexpress.com/item/1005007343441298.html#nav-specification) |  |
| Temperatuursensor | [Sht30 Sht31 Sht35 Hoge Precisie Waterdichte Hoge Temperatuur-Resistente Luchttemperatuur Vochtigheid Meting I2c Digitale Sensor - AliExpress 30](https://nl.aliexpress.com/item/1005006339627671.html?src=bing&aff_short_key=UneMJZVf&aff_platform=true&isdl=y&albch=shopping&acnt=135095331&isdl=y&albcp=554602608&albag=1295225801673120&slnk=&trgt=pla-4584551179236980&plac=&crea=80951665855673&netw=o&device=c&mtctp=e&utm_source=Bing&utm_medium=shopping&utm_campaign=PA_Bing_NL_PLA_PC_SFFC_0-7_ECPC_20230825&utm_content=customlable1%3D7&utm_term=SENSIRION%20SHT35&msclkid=33ce4ae03279164db09d2d4a006ffe1d&gatewayAdapt=glo2nld) |  |
| Deur sensor | (geen datasheet) volgens het internet zitten er 4 draden op. 1, + 24 volt. Deze wordt doorverbonden naar de computer als dicht | Volt -v  Naar input |
| Toerentalsensor | 240 volt via i2c/bus. | 240volt / kan naar i2c |

De benodigdheden per product en aansluiting wordt weergeven sommige componenten moeten via een alternatieve besturing worden aangesloten.

# Besturingssysteem en Software

## Besturingssysteem van de PLC

PLC’s gebruiken geen traditioneel besturingssysteem zoals Windows of Linux. In plaats daarvan gebruiken ze vaak een embedded real-time besturingssysteem (RTOS). Dit besturingssysteem is geoptimaliseerd voor het snel en betrouwbaar uitvoeren van de programma's die de PLC aansturen.

Het besturingssysteem zorgt ervoor dat de PLC snel input kan verwerken, de juiste output kan sturen, en dat de timing en logica nauwkeurig zijn, wat essentieel is voor industriële processen.

**Voorbeelden van embedded besturingssystemen in PLC's:**

Siemens: gebruikt een eigen systeem in hun PLC’s zoals de S7-serie.

Schneider Electric: biedt Modicon PLC’s die ook op een embedded besturingssysteem draaien.

**Programmeeromgeving:**

Ondersteunt verschillende programmeertalen zoals Ladder Logic (LAD), Structured Text (ST), Function Block Diagram (FBD) en Sequential Function Chart (SFC), allemaal volgens de IEC 61131-3 standaard.

**communicatieprotocollen**

volgens het internet kwam dit.

PLC’s gebruiken verschillende communicatieprotocollen voor het verbinden van de PLC met andere apparaten, systemen, en netwerken:

Profinet / Ethernet/IP (voor Siemens en Allen-Bradley)

Modbus-TCP / Modbus RTU (voor Schneider en Beckhoff)

CANopen, EtherCAT, OPC UA voor communicatie met veldapparatuur, sensoren, en andere systemen.

Besturingssysteem van de Arduino Mega.  
er is niet een specifiek besturing systeem voor de Arduino. Ook zijn er geen oplsag mogelijkheden in de standaard module. Dit betekend dat na een tijd de arduino meer gaan vastlopen als we gebruik maken van een klok. We moeten dus hier ook een speciale klok unit.

De Arduino mega heeft wel de mogelijkheid voor RTOS, dit is Real-Time Operating System. Dit is optioneel. Hierom ga ik hier geen gebruik van maken voor dit project.

**Programmeeromgeving:**

De Arduino is net als alle andere microcontrollers. we kunnen ze allemaal vaak programmeren met C/C++. We hebben hier het voordeel dat er veel voorbeeld matriaal is en vooral als we C++ gebruiken zijn er veel bibliotheken.

**Communicatieprotocollen**

volgens het internet hebben bijna alle microcontrollers de volgende Communicatieprotocollen

**I2C (Inter-Integrated Circuit):**

De Arduino ondersteunt I2C via de Wire Library. Dit maakt het eenvoudig om meerdere sensoren of apparaten aan te sluiten met slechts twee draden (SCL en SDA).

**SPI (Serial Peripheral Interface):**

SPI is een veelgebruikte communicatie-interface voor snel dataoverdracht. De SPI Library stelt je in staat om communicatie tot stand te brengen met SPI-compatibele apparaten, zoals geheugenkaarten, displays en sensoren.

**UART (Universal Asynchronous Receiver Transmitter):**

Via de Serial Library kun je seriële communicatie opzetten met andere apparaten, zoals een computer of ander microcontrollerbord.

**PWM (Pulse Width Modulation):**

De Arduino Mega ondersteunt PWM-uitgangen op verschillende pinnen, die je kunt gebruiken voor het aansteken van LEDs, het aansteken van motoren of het regelen van de snelheid van apparaten.

**Ethernet / Wi-Fi / Bluetooth:**

Je kunt modules zoals de Ethernet Shield, Wi-Fi Shield (zoals de ESP8266 of ESP32) of Bluetooth modules (HC-05, HC-06) gebruiken om netwerkcommunicatie tot stand te brengen.

**Modbus-TCP / Modbus RTU:**

De Arduino ondersteunt Modbus communicatie via bibliotheken zoals ModbusMaster (voor Modbus RTU) en EthernetModbus (voor Modbus TCP). Dit maakt het mogelijk om je Arduino in een industriële setting te integreren met andere Modbus-compatibele apparaten.

## Besturingssysteem van de Rasberry pi 3

De rasberry pi 3 maakt gebruik van een cpu. Dit is verschillende per versie en leverancier. De rasberry pi 3 is

CPU: ARM Cortex-A53

Aantal cores: 4 cores

Kloksnelheid: 1.2 GHz

Het voordeel is dat met keuze is dat het mogelijk is om in alle talen te programmeren.

**Programmeeromgeving:**

De **Raspberry Pi 3** biedt veelzijdige programmeermogelijkheden, zowel voor beginners als voor gevorderde gebruikers. Dit maakt de Raspberry Pi aantrekkelijk voor prototyping, educatie en zelfs industriële toepassingen.

**A. Python**

* **Python** is de meest populaire taal voor de Raspberry Pi, met name vanwege de eenvoud en de enorme hoeveelheid beschikbare bibliotheken.
  + **GPIO**-besturing wordt mogelijk gemaakt door de **RPi.GPIO** bibliotheek.
  + De **pigpio**-bibliotheek biedt een krachtige manier om GPIO-pinnen aan te sturen met precisie.

**B. C / C++**

* **C/C++** is een uitstekende keuze voor het schrijven van high-performance toepassingen en is handig als je meer controle nodig hebt over hardware.
  + De **WiringPi** bibliotheek was een populaire keuze voor het werken met GPIO-pinnen in C, hoewel deze nu niet meer wordt onderhouden. Alternatieven zoals **pigpio** en **wiringPi-next** zijn echter beschikbaar.

**C. JavaScript (Node.js)**

* **Node.js** en **Johnny-Five** zijn uitstekende tools voor het bouwen van robotica- of automatiseringsprojecten met JavaScript.
  + **Node-RED** is een flow-gebaseerde ontwikkeltool waarmee je visueel kunt programmeren door blokken van code te verbinden, en het ondersteunt veel protocollen voor automatisering (zoals MQTT, Modbus, en meer).

**D. Shell-scripting (Bash)**

* Omdat de Raspberry Pi draait op Linux, kun je ook gebruik maken van **Bash scripting** om eenvoudigere automatiseringsscripts te schrijven voor het aansteken van apparaten, uitvoeren van netwerkcommunicatie en het beheren van besturingssystemen.

**E. C# / .NET**

* Voor degenen die werken in een **Windows 10 IoT Core** omgeving of die liever met C# werken, biedt de Raspberry Pi ook **.NET Core** ondersteuning.
  + **Windows IoT Core** kan een geschikte keuze zijn voor industriële toepassingen als je al werkt met de Microsoft-stack.

**communicatieprotocollen**

De **Raspberry Pi 3** biedt een breed scala aan communicatieprotocollen, waardoor het ideaal is voor het verbinden van sensoren, actuatoren en andere apparaten in een netwerk.

**Communicatieprotocollen en Integratie**

De **Raspberry Pi 3** biedt een breed scala aan communicatieprotocollen, waardoor het ideaal is voor het verbinden van sensoren, actuatoren en andere apparaten in een netwerk.

**A. GPIO (General Purpose Input/Output)**

* De **Raspberry Pi 3** heeft **40 GPIO-pinnen**, die kunnen worden geconfigureerd als digitale ingangen of uitgangen. Deze pinnen kunnen direct worden aangestuurd via Python (RPi.GPIO, pigpio) of C/C++.

**B. I2C (Inter-Integrated Circuit)**

* De **I2C-interface** is beschikbaar via de **wiringPi** en **pigpio** bibliotheken. Het ondersteunt communicatie met meerdere apparaten (bijv. sensoren, displays, enz.) via slechts twee draden (SDA en SCL).

**C. SPI (Serial Peripheral Interface)**

* De **SPI-interface** kan worden gebruikt om te communiceren met apparaten zoals displays, motoren en geheugenmodules via de **wiringPi** of **pigpio** bibliotheken.

**D. UART (Universal Asynchronous Receiver/Transmitter)**

* De **Raspberry Pi 3** biedt **UART**-ondersteuning via GPIO-pinnen en kan communiceren met andere apparaten zoals microcontrollers, GPS-modules en sensoren. Dit kan via de **Serial** module in Python of via **wiringPi** in C.

**E. Ethernet / Wi-Fi / Bluetooth**

* De **Raspberry Pi 3** heeft ingebouwde **Wi-Fi** en **Bluetooth**-ondersteuning, wat betekent dat je eenvoudig netwerkcommunicatie kunt opzetten met andere apparaten via **Ethernet**, **Wi-Fi**, of **Bluetooth**.
  + **MQTT** is een populair protocol voor machine-to-machine (M2M) communicatie en wordt vaak gebruikt voor Internet of Things (IoT) toepassingen. Je kunt MQTT-communicatie opzetten via Python, JavaScript (Node.js), of C/C++.

**F. Modbus**

* **Modbus**-communicatie is ook beschikbaar op de Raspberry Pi via verschillende bibliotheken, zoals **pyModbus** (voor Python) en **libmodbus** (voor C). Dit maakt het mogelijk om met industriële apparaten en sensoren te communiceren.

**G. Web-communicatie en HMI**

* De Raspberry Pi kan eenvoudig een **webserver** hosten, bijvoorbeeld met **Apache** of **Nginx**, en een **HMI (Human-Machine Interface)** bieden via een webinterface.
  + **Node-RED** kan worden gebruikt voor het visueel ontwerpen van HMI’s en het aansteken van sensoren of actuatoren via een browser.
  + **Flask** (voor Python) of **Express** (voor Node.js) kunnen ook worden gebruikt voor het ontwikkelen van webgebaseerde interfaces.

**HMI / SCADA Software (Visualisatie)**

De Raspberry Pi kan eenvoudig worden gebruikt om een HMI (Human Machine Interface) te creëren voor de visualisatie en bediening van het systeem.

1. **Node-RED**:
   * **Node-RED** is een visuele ontwikkeltool die draait op de Raspberry Pi en ideaal is voor het creëren van flow-gebaseerde HMI’s. Het ondersteunt grafische interfaces voor het besturen van apparaten via MQTT, Modbus, en meer.
2. **Grafana / InfluxDB**:
   * **Grafana** en **InfluxDB** kunnen worden geïnstalleerd om gegevens te verzamelen en weer te geven in dashboards. Dit wordt vaak gebruikt voor monitoring en visualisatie van data in een SCADA-systeem.
3. **Web-based HMI**:
   * Je kunt zelf een **webinterface** ontwikkelen met behulp van HTML, CSS, en JavaScript. Frameworks zoals **Flask (Python)** of **Express (Node.js)** kunnen een server draaien die je via de browser kunt bedienen.

## Besturingssysteem van een zelfontwikkelde printplaats

Er is geen mogelijkheid voor programmering . het besturings systeem is voornamelijk logica

**Programmeeromgeving:**

Er zijn geen programmeer systeem echter zijn er wel programma’s waarbij je kan simuleren wat alles gaat doen HDL Designer is een voorbeeld

**communicatieprotocollen**

niet van toepassing.

# voor en nadelen.

plc

**Voordelen:**

Robuustheid: Geschikt voor industriële omgevingen, bestand tegen zware omstandigheden zoals stof, vocht en extreme temperaturen.

Real-time besturing: Zeer geschikt voor processen die snel reageren op inputs (zoals in fabrieken, machines en processen).

Betrouwbaarheid: Zeer betrouwbaar voor taken waarbij geen storing mag optreden, zoals bij de controle van productielijnen.

Gebruiksvriendelijke software: Meestal wordt er gewerkt met een grafische programmeertaal zoals Ladder Logic (LD), Function Block Diagram (FBD), of Structured Text (ST), wat het eenvoudig maakt voor technici zonder diepgaande programmeerkennis.

**Nadelen:**

Hoge kosten: PLC's en de bijbehorende softwarelicenties kunnen duur zijn, wat ze minder geschikt maakt voor kleine projecten.

Beperktere programmeermogelijkheden: De programmeertalen zijn vaak minder flexibel dan bij een algemene computer, en je hebt beperkte mogelijkheden om complexe algoritmes of grafische toepassingen uit te voeren.

arduino

**Voordelen:**

Kosten: Zeer goedkoop in vergelijking met PLC's.

Eenvoudige programmeerbare interface: Programmeert met Arduino IDE, wat het eenvoudig maakt om snel code te schrijven en te testen.

Grote community: Veel ondersteunende forums, bibliotheken en voorbeelden beschikbaar voor snelle ontwikkeling.

Gebruik van hardware: Directe toegang tot GPIO-pinnen maakt het eenvoudig om sensoren, motoren en andere hardware aan te sturen.

**Nadelen:**

Beperkingen qua verwerkingskracht: De ATmega2560 microcontroller op de Arduino Mega heeft beperkte rekenkracht en geheugen. Het is niet geschikt voor zware reken- of grafische toepassingen.

Niet geschikt voor complexe besturingssystemen: Dit is geen besturingssysteem zoals Linux of Windows, dus als je meer geavanceerde functionaliteit nodig hebt, zoals netwerken of uitgebreide geheugenbehoeften, kan de Arduino tekortschieten.

rasberry 3pi

**Voordelen:**

Krachtige CPU: De ARM Cortex-A53 CPU van de Raspberry Pi 3 heeft meer rekenkracht dan de meeste microcontrollers en PLC's.

Flexibele programmeermogelijkheden: Kan vrijwel elke programmeertaal uitvoeren, zoals Python, C++, Java, Node.js, enzovoort.

Volledig besturingssysteem: Het draait op een besturingssysteem zoals Raspberry Pi OS (Linux-gebaseerd), wat het mogelijk maakt om complexe toepassingen te ontwikkelen.

Netwerkfunctionaliteit: De Pi heeft ingebouwde WiFi en Ethernet, waardoor het perfect is voor Internet of Things (IoT)-projecten.

Multimedia en grafische mogelijkheden: Kan werken met grafische interfaces en biedt uitgebreide mogelijkheden voor visualisatie en besturingsinterfaces.

**Nadelen:**

Niet real-time: Het draait op een besturingssysteem dat geen real-time prestaties garandeert, wat problematisch kan zijn voor industriële besturingen.

Hogere energieconsumptie: De Raspberry Pi verbruikt meer energie dan een Arduino of PLC, wat in bepaalde toepassingen een nadeel kan zijn.

Complexiteit: Het besturingssysteem en de programmeermogelijkheden kunnen overkill zijn voor eenvoudige taken, en je hebt meer kennis van Linux en programmeren nodig.

Zelfontwerp

**Voordelen:**

Maatwerk: Je kunt de printplaat volledig aanpassen aan de behoeften van je project. Je kunt de keuze maken voor de microcontroller, pinout, sensoren, actuatoren en andere functies die je nodig hebt.

Flexibiliteit in componenten: Je hebt de volledige controle over welke componenten je gebruikt, wat handig is voor toepassingen die specifieke eisen hebben.

Kosteneffectief bij grootschalige productie: Als je van plan bent om je ontwerp op grotere schaal te produceren, kan een op maat gemaakte printplaat goedkoper zijn dan het gebruik van standaard platforms zoals een Arduino of Raspberry Pi.

Geen afhankelijkheid van externe hardware: Je bent niet afhankelijk van commerciële oplossingen. Dit kan nuttig zijn als je een specifieke integratie of ontwerp nodig hebt.

Betere controle over energieverbruik: Je kunt componenten kiezen die energiezuinig zijn, wat handig is voor toepassingen waar batterijduur belangrijk is.

**Nadelen:**

Complexiteit in ontwerp: Het ontwerpen van een PCB vereist gedegen kennis van elektronica en PCB-layout. Je moet in staat zijn om de schema's en de lay-out goed te plannen.

Tijd en kosten voor ontwerp en fabricage: Het ontwerpen en fabriceren van een PCB kan tijdrovend zijn en initiële kosten met zich meebrengen. Je moet ook zorgen voor kwaliteitscontrole tijdens het productieproces.

Geen kant-en-klare software: In tegenstelling tot platformen zoals Arduino, Raspberry Pi, of PLC's, moet je de software en firmware helemaal zelf ontwikkelen, wat extra tijd en expertise vereist.

Debuggen: Het kan moeilijker zijn om bugs te vinden als je alles zelf ontwerpt, omdat er geen standaard ontwikkelingsomgeving is die alle problemen automatisch kan identificeren.

Om alle gegevens samen te vatten heb ik een tabel gemaakt

# Voordeel en nadeel in tabel

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Kenmerk | PLC | Arduino Mega | Raspberry Pi 3 | Microcontroller (STM32, etc.) | Zelf Ontworpen PCB |
| Kosten | Hoog | Laag | Laag | Laag tot gemiddeld | Afhankelijk van ontwerp (vaak laag bij grotere aantallen) |
| Rekenkracht | Beperkt (voor basisbesturing) | Beperkt (8-bits, 16 MHz) | Hoog (Quad-core ARM Cortex-A53) | Middelmatig tot hoog (afhankelijk van type) | Afhankelijk van gekozen microcontroller (kan variëren) |
| Programmeertaal | Grafisch (Ladder Logic, FBD, ST) | C, C++, Arduino IDE | Python, C++, Java, Node.js, etc. | C, C++ (meestal bare-metal) | C, C++, Assembly, afhankelijk van de gekozen microcontroller |
| Gebruiksgemak | Eenvoudig, maar beperkt | Zeer eenvoudig met IDE en bibliotheken | Complexer (Linux OS, command-line) | Complexer, vereist meer hardwarekennis | Complexer, afhankelijk van ontwerp en tools |
| Toepassingen | Industriële automatisering | Prototyping, hobbyprojecten | IoT, computer vision, webservers | Embedded systemen, real-time toepassingen | Maatwerkoplossingen, zeer specifieke toepassingen |
| Netwerkfunctionaliteit | Beperkt (indien ondersteund) | Via externe modules (Wi-Fi, Ethernet) | Ingebouwd Wi-Fi en Ethernet | Vaak via externe modules | Afhankelijk van ontwerp (kan Wi-Fi, Bluetooth, etc. omvatten) |
| Real-time capaciteiten | Uitstekend | Beperkt (niet real-time) | Beperkt (niet real-time) | Goed (afhankelijk van configuratie) | Kan goed zijn (real-time besturing mogelijk, afhankelijk van keuze microcontroller) |
| Complexiteit | Laag (voor basisbesturing) | Laag | Hoog (voor gevorderde projecten) | Hoog (voor embedded en real-time systemen) | Hoog (vereist volledig ontwerp en firmwareontwikkeling) |
| Energieverbruik | Laag (geoptimaliseerd voor industriële toepassingen) | Laag | Middelmatig (kan relatief veel verbruiken) | Laag (meestal energie-efficiënt) | Zeer laag (afhankelijk van ontwerp) |
| Maatwerk/Integratie | Beperkt tot fabrikantenkeuzes | Beperkt (gebruik van modules) | Beperkt (beperkingen door Raspberry Pi ontwerp) | Hoog (volledige controle over ontwerp) | Zeer hoog (volledig aanpasbaar) |
| Kenmerk | PLC | Arduino Mega | Raspberry Pi 3 | Microcontroller (STM32, etc.) | Zelf Ontworpen PCB |
| Kosten | Hoog | Laag | Laag | Laag tot gemiddeld | Afhankelijk van ontwerp (vaak laag bij grotere aantallen) |
| Rekenkracht | Beperkt (voor basisbesturing) | Beperkt (8-bits, 16 MHz) | Hoog (Quad-core ARM Cortex-A53) | Middelmatig tot hoog (afhankelijk van type) | Afhankelijk van gekozen microcontroller (kan variëren) |
| Programmeertaal | Grafisch (Ladder Logic, FBD, ST) | C, C++, Arduino IDE | Python, C++, Java, Node.js, etc. | C, C++ (meestal bare-metal) | C, C++, Assembly, afhankelijk van de gekozen microcontroller |