OS: Application

Joran van Teeffelen

Inleiding

In deze opdracht worden alle eerder gemaakte opdrachten gecombineerd tot 1 programma. Hierin maak je een gamepad besturingsprogramma dat vanuit een browser aan te roepen is.

Opdracht

Bouw het gamepad besturingsprogramma wat je in een eerdere opdracht hebt gerealiseerd om tot een daemon – een service – die op de achtergrond draait. Daarnaast dient een losse CGI-applicatie geschreven te worden, die vanuit een remote browser via het netwerk kan worden aangeroepen, en die door middel van IPC mechanismes, met name POSIX MessageQueue en SharedMemory, communiceert met deze gamepad daemon.

De applicatie moet (‘*must’*) voldoen aan de volgende eisen:

- Algemeen:
  - Zelf met BuildRoot geconfigureerd en gebouwd Embedded Linux image
  - Applicatie start automatisch als het systeem wordt ge-reboot (en bij power-on)
- Knoppen inlezen:
  - Daemon publiceert de toestand van de gamepad knoppen in POSIX SharedMemory.
  - CGI-applicatie leest shared memory op verzoek van de browser.
- LEDjes en Rumble aansturen:
  - CGI-app vertaalt browser commando in message in een POSIX MessageQueue.
  - Daemon haalt commando uit de MessageQueue en stuurt gamepad aan.

Het volgende zou eigenlijk ook moeten (‘*should’*):

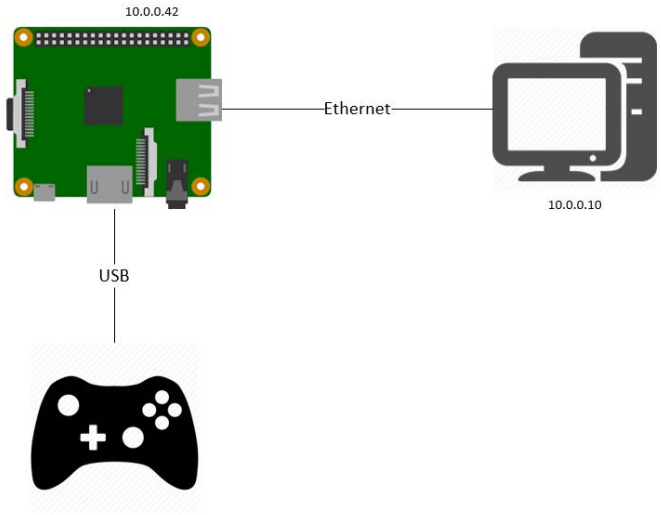
- 2 (of meer) gamepads worden ondersteund, bij voorkeur hot-pluggable

Hoe een en andere zichtbaar gemaakt wordt in de browser wordt aan de student overgelaten. Dit is een leuke aanleiding om je in JavaScript te verdiepen (maar is niet verplicht). Mooi zou zijn (‘*could’*):

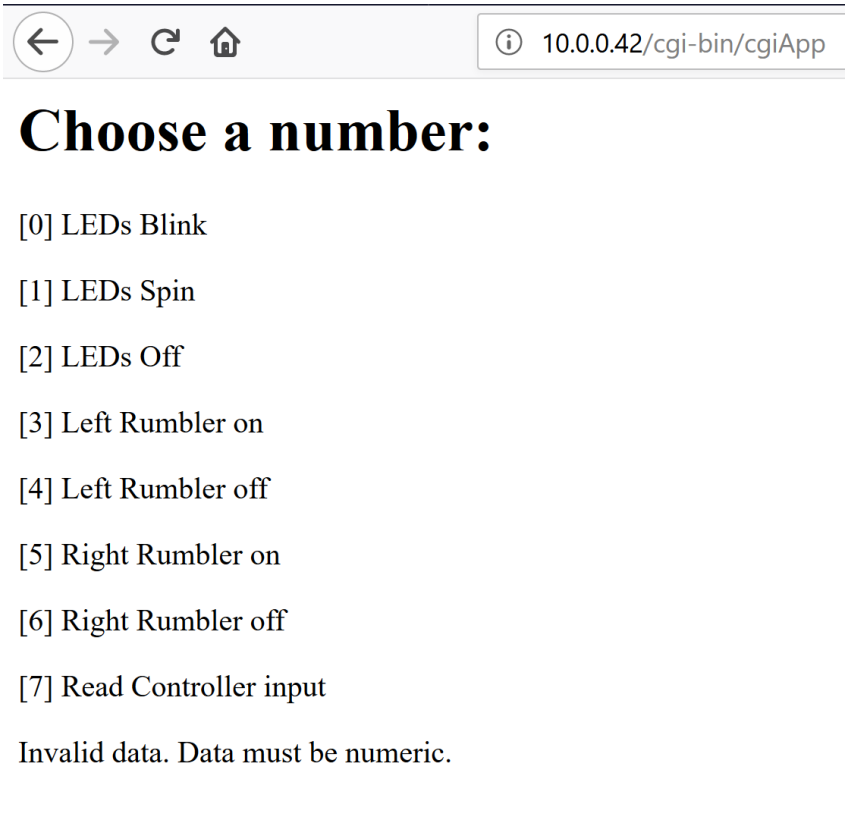
- Grafische user-interface in de browser.

Werking

De opstelling ziet er als volgt uit:

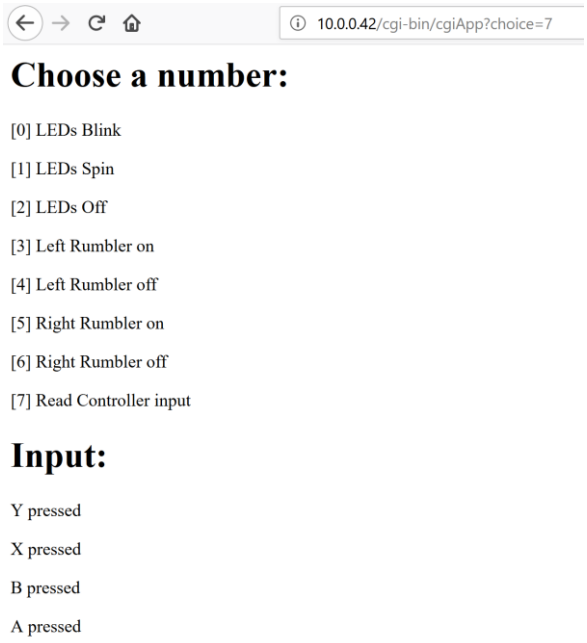


Hierin is te zien dat de controller is aangesloten via USB aan de Raspberry Pi. De Pi is verbonden met de computer via ethernet om te communiceren. Op de computer kun je vanuit een browser de CGI uitvoeren. Dit kun je doen door naar *10.0.0.42/cgi-bin/cgiApp* te navigeren.



Hierin zie je een overzicht van alle cijfers die je kan invoeren en de corresponderende acties.

Een actie kun je invoeren door achter de huidige link *?choice=[cijfer]* in te voeren. Als je 1 t/m 6 invult, zie je het resultaat op de controller en als je 7 invult zal je het resultaat onder de lijst zien.



Als er een keuze is ingevoerd, kijkt de CGI eerst in de environment list te zoeken naar “QUERY_STRING”. Dit correspondeert met variabelen die in de URL worden geschreven.

Vervolgens kijkt het programma welke keuze is gemaakt en activeert, bij keuze 7, de reading methode en bij de andere keuzes de handleInput methode.

De reading methode opent eerst de semaforen en het gedeelde geheugen en controleert of deze wel correct geopend zijn. Vervolgens wordt er een Post geschreven naar de requestSemafoor. Hierin vraagt de CGI om een nieuw report te maken van de controllerconfiguratie en deze naar het geheugen te schrijven. Hierna wacht het programma op de daemon tot ie klaar is met schrijven en een post stuurt naar de availableSemafoor. Vervolgens kan het programma het stukje geheugen uitlezen en casten naar de struct die correspondeert met de controllerconfiguratie. En als laatste worden de semaforen en het shared memory afgesloten.

De handleInput methode opent de bestaande message queue (omdat de daemon eerst een message queue aan moet maken wanneer deze opgestart is) en schrijft het ingevoerde getal naar het geheugen. Ondertussen wacht de daemon op messages en als er een gevonden is, gaat hij een switch af voor de waarde en stuurt de corresponderende actie naar de controller.