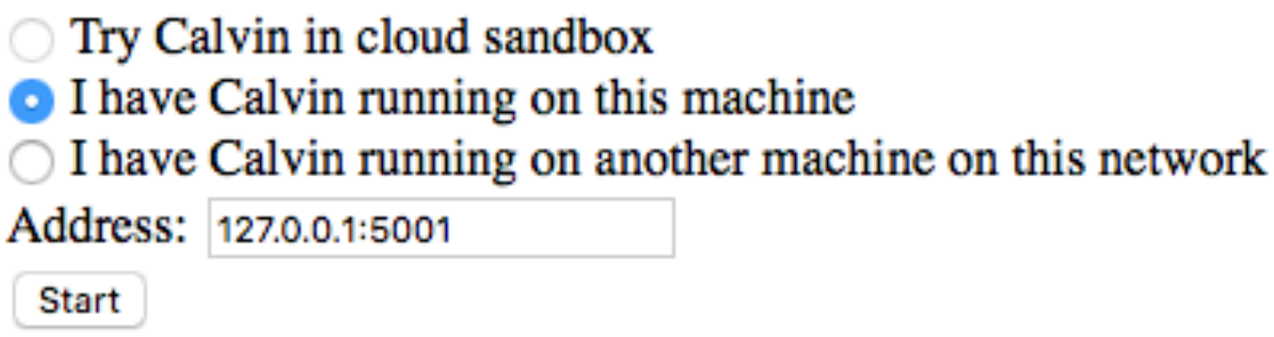
Bygg en applikation med CalvinGUI

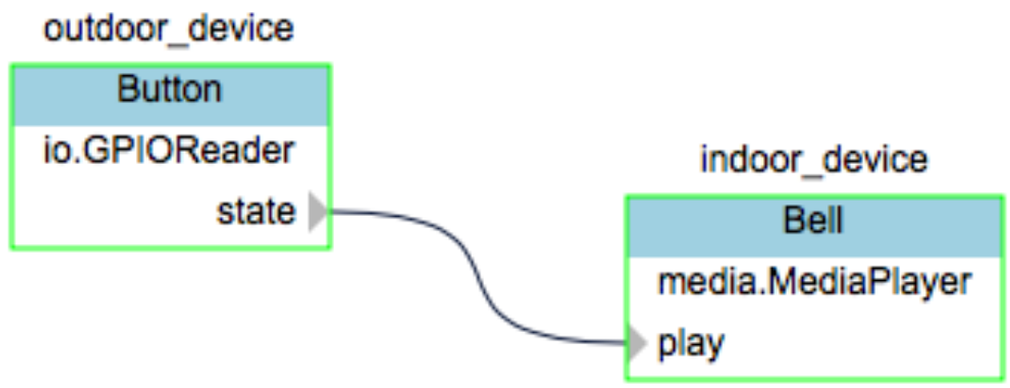
- Denna guide förutsätter att scripten *part_1.calvin* och *part_2.calvin* har körts igenom med förväntat resultat.

- Starta tre runtime, en på datorn och en på varje Raspberry Pi, använd startscripten i exemplet:
    - ./start-computer.sh 127.0.0.1
    - ./start-outdoor.sh <ip address>
    - ./start-indoor.sh <ip address>
- Gå till address: **http://94.246.117.105** i en webbrowser.
- Välj “**I have Calvin running on this machine**” och adress: **127.0.0.1:5001**
- Tryck på **Start**



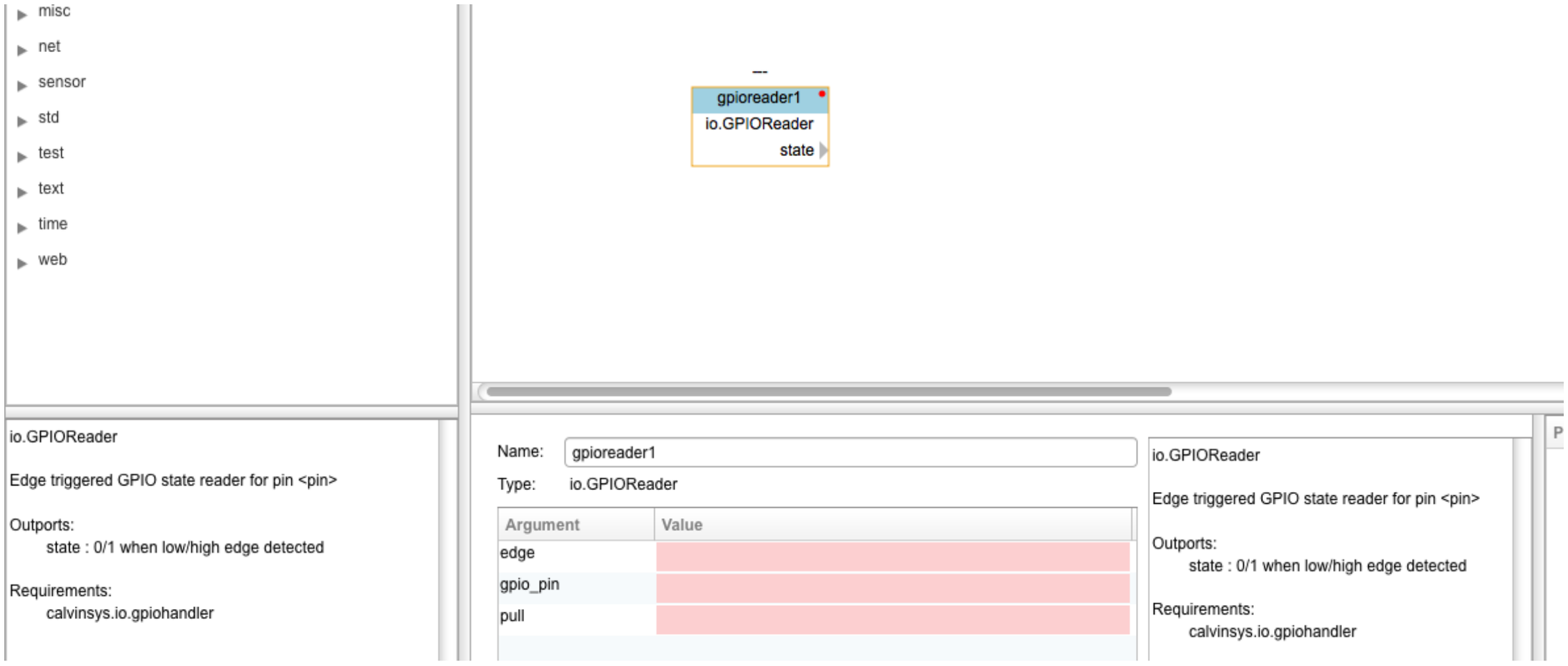
Ringklocka

- Följande exempel visar hur en applikation för en ringklocka kan byggas i Calvin GUI.

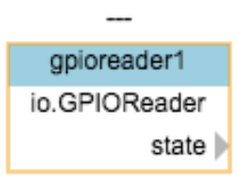


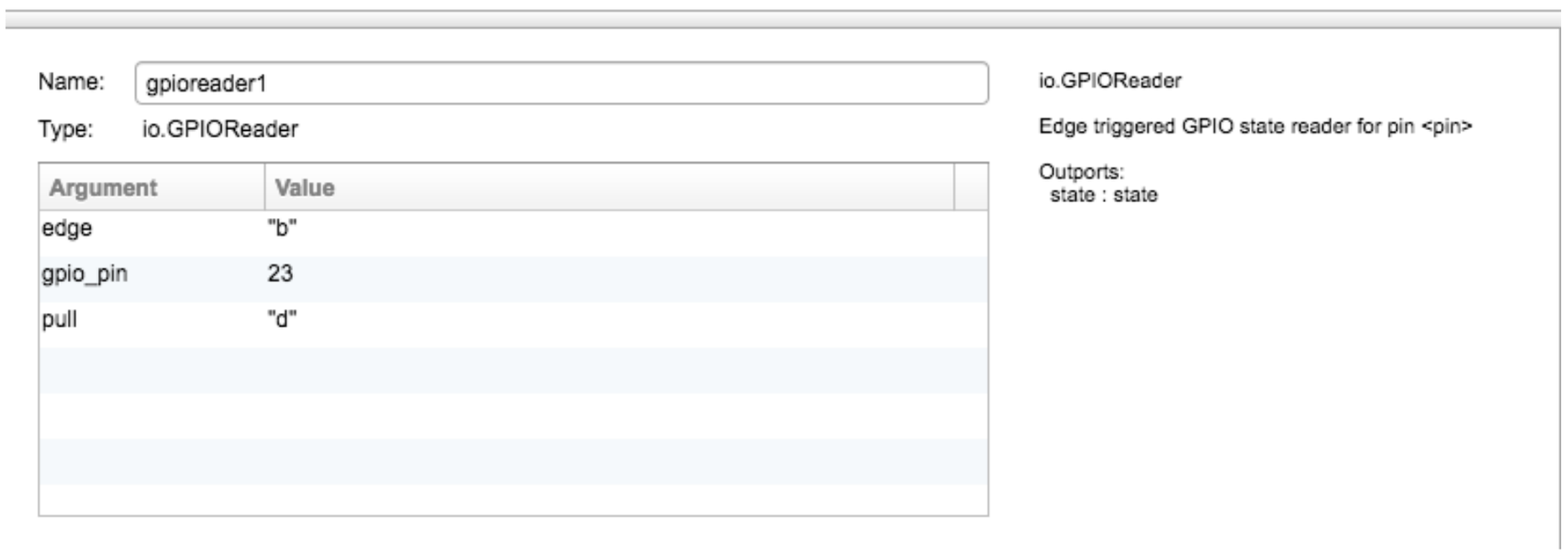
- Välj aktören GPIOReader under kategorin **io,** drag ut och släpp den i mittersta fönstret.

- I den nedre mittersta delen av fönstret är argumenten för att initiera aktören synliga. Röda fält indikerar att värden måste anges för att kunna initiera aktören.



- Initiera aktören med följande värde
- edge=**“b”**
- gpio_pin=**23**
- pull=**“d”**
- För att applikationen ska bli lättare att överblicka är det även en god idé att namnge aktören till något annat än gpioreader1. Välj ett lämpligt namn för aktören och fyll i fältet Name.

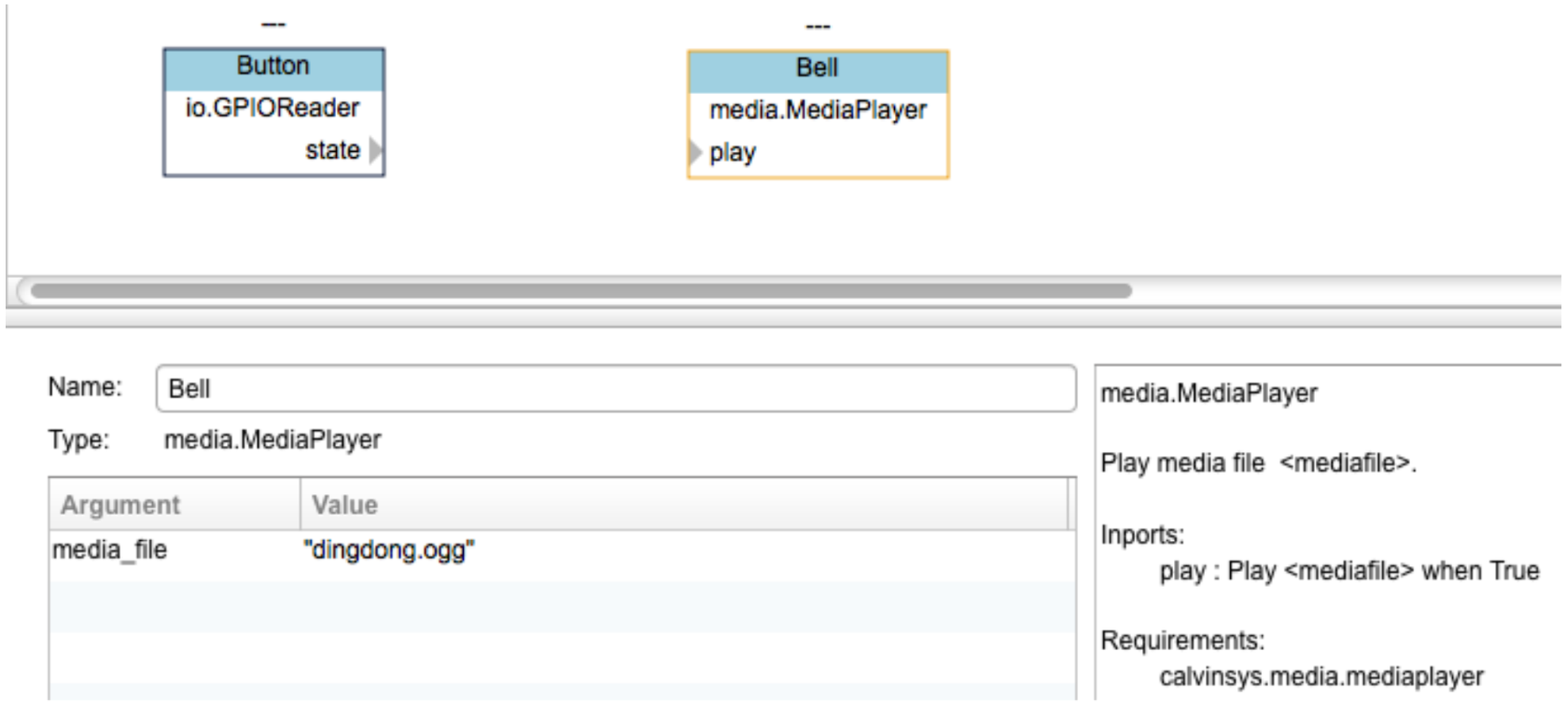




- I ActorStore under kategorin **media** finns en MediaPlayer aktör. Dra ut och släpp den intill GPIOReader aktören.

- MediaPlayer behöver initieras med adressen till en ljudfil. I exemplet ingår en fil *<dingdong.ogg>* som fungerar bra för en dörrklocka.

- Namnge även denna aktör.



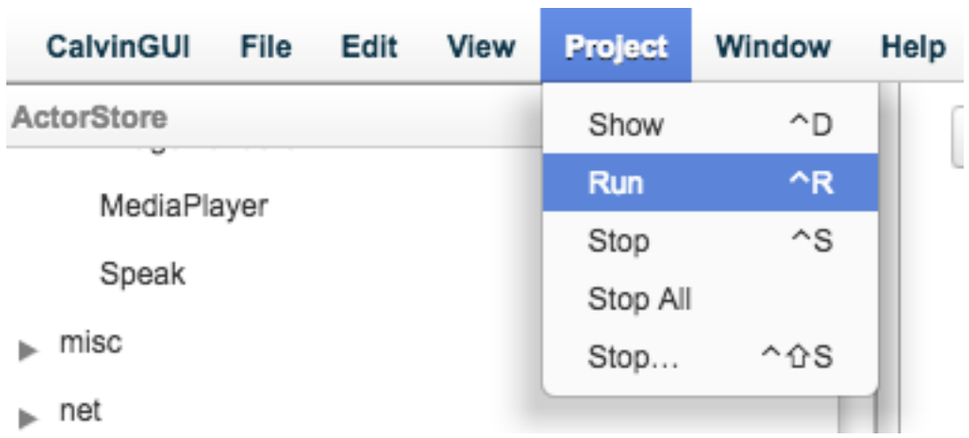
- Nu är applikationen nästan färdigt, men för att data ska kunna skickas mellan aktörerna måste portarna kopplas ihop. Dessa representeras av de små grå pilarna i varje aktör.

- Klicka på GPIOReader aktörens ut-port och dra en linje till in-porten på MediaPlayer aktören för att skapa en koppling mellan portarna.

- GPIOReader aktören kommer skicka **0** på sin ut-port **state** när den känner av att knappen släpps upp och **1** när den trycks ner.

- När MediaPlayer aktören får ett token som kan evakueras till True, tex **1**, på sin in-port **play**, kommer aktören att spela upp sin media fil.

- Starta applikationen genom att välja **Run** i menyn **Project**

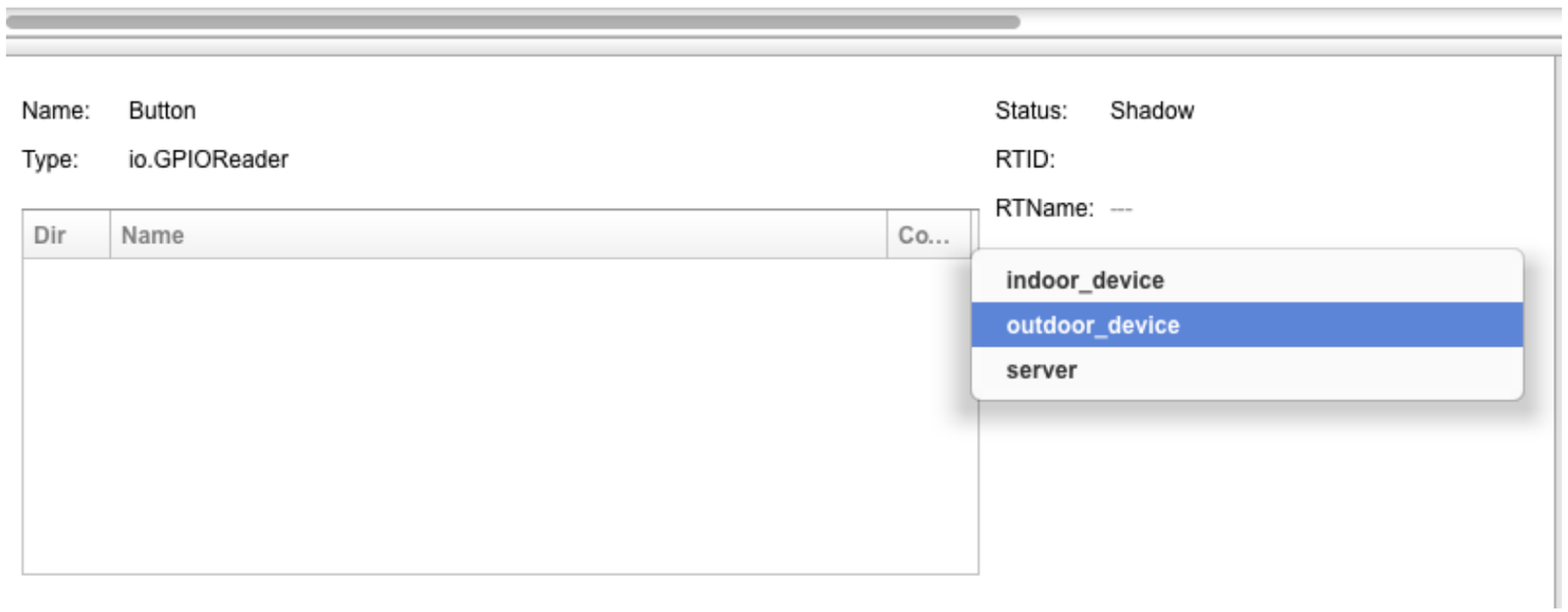


- Applikationen kör nu men ingen av aktörerna är kapabla att utföra sina uppgifter. Detta beror på att config filen för runtime *server* inte specificerar någon gpio- eller media-plugin. Båda aktörerna är därför skuggaktörer (shadow actors).

- Att aktörerna är skuggaktörer är i nuläget inte synligt i Calvin GUI. För att verifiera att så är fallet är ett alternativ att använda verktyget **csweb**.

- För att knappen ska fungera måste aktören GPIOReader flyttas till en Raspberry Pi med en fysisk knapp.

- Markera aktören GPIOReader och välj **outdoor_device** i runtimemenyn. Aktören kommer migreras till den Rasperry Pi med en runtime som heter *outdoor_device*.



- Migrera MediaPlayer på samma vis, men denna gång till runtime **indoor_device**.

- Om migreringen gick bra ska texten över aktörerna uppdateras, klicka en gång i fönstret för att uppdatera texten om det inte sker automatiskt.

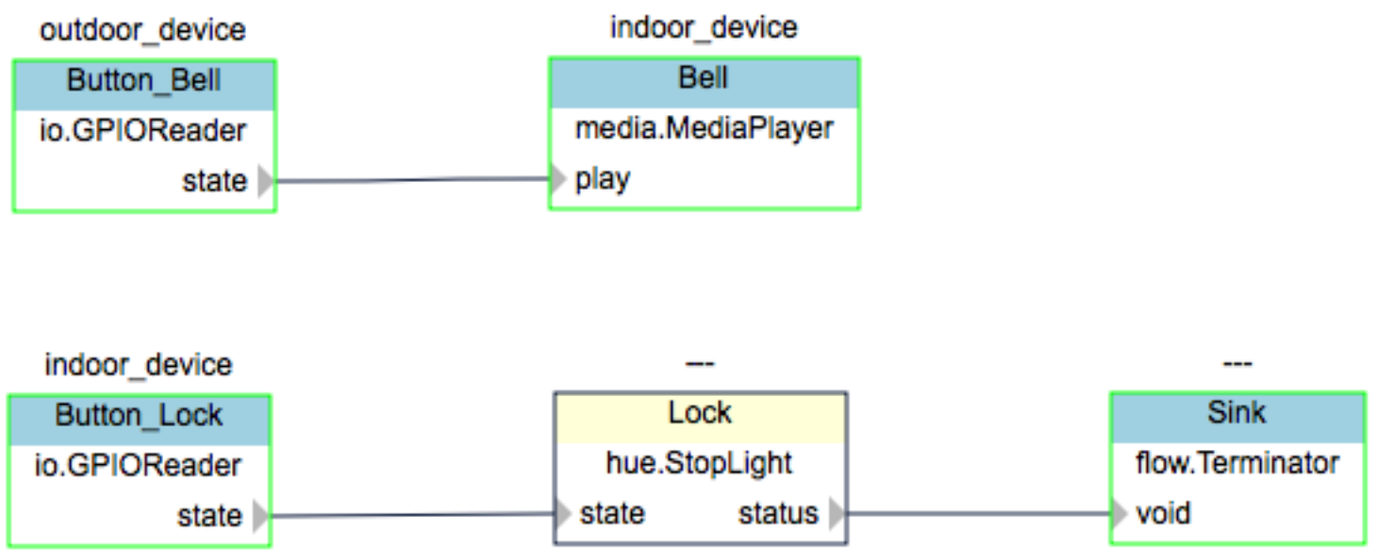
- Testa om applikationen fungerar genom att trycka på knappen kopplad till den Raspberry Pi dit GPIOReader aktören migrerat.

- Om inget händer finns några debugtips i slutet av denna manual.

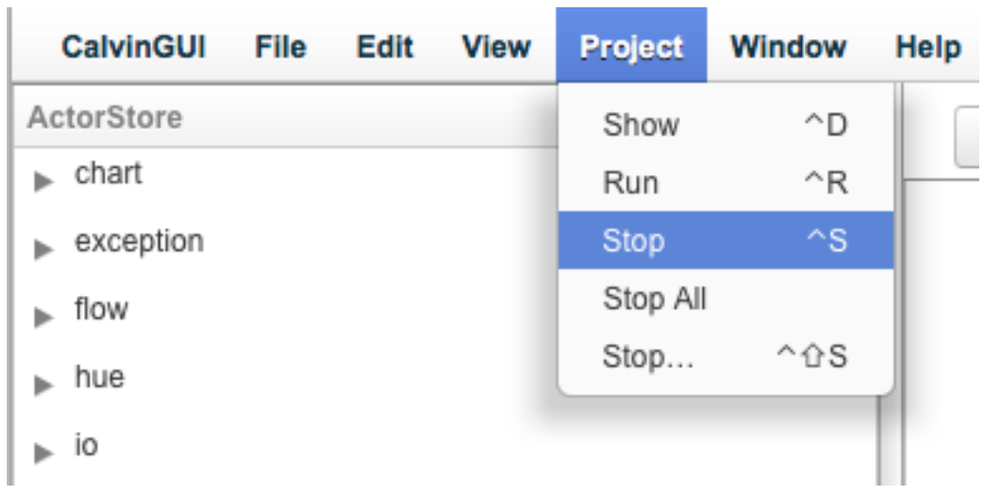
Lås

- Applikationen ska nu byggas ut med en Philips Hue lampa som kan styras från insidan. Den kommer visa grönt ljus för att indikera att dörren öppnas. Rött ljus betyder låst.

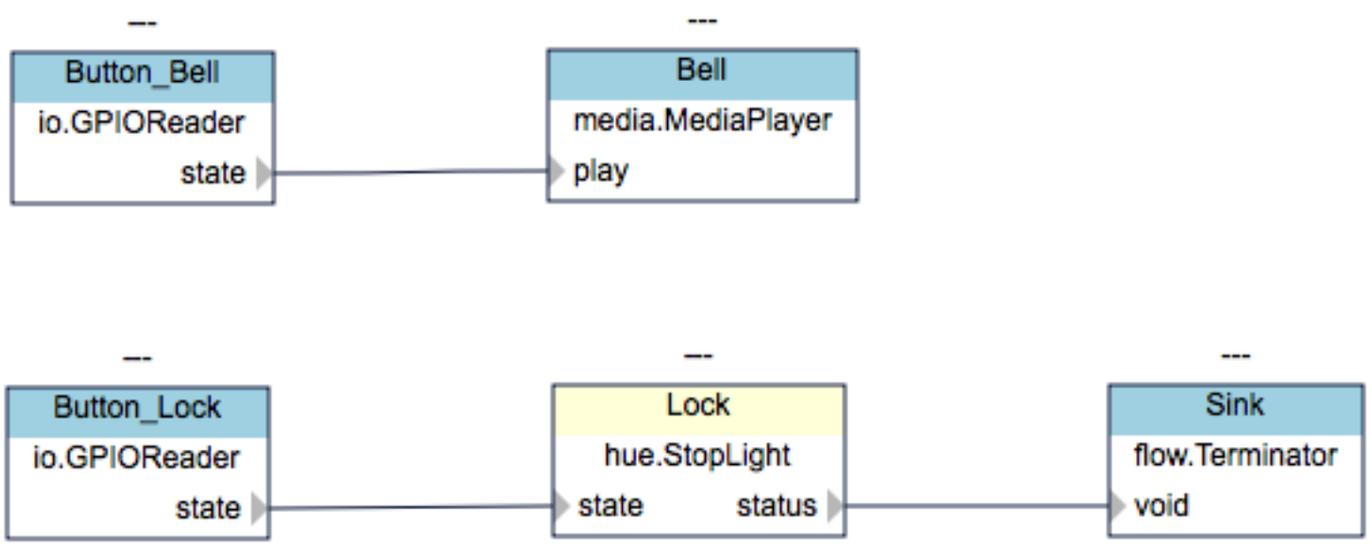
- Om ingen Philips Hue är tillgänglig kan terminalen användas för att visa när dörren öppnas istället. Hoppa isåfall till **Lås utan Hue** sektionen.



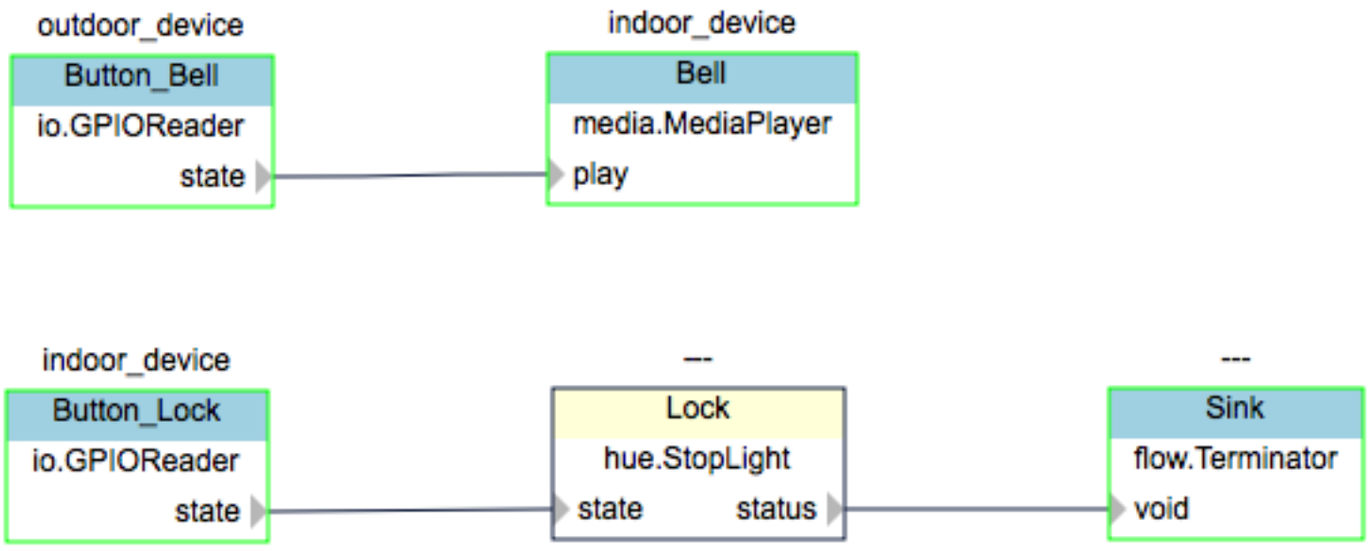
- Börja med att stoppa förra applikationen genom att välja **Stop** i menyn **Project**



- Utöver aktörerna GPIOReader och MediaPlayer, dra ut följande nya aktörer till arbetsfönstret.
    - GPIOReader under kategorin **io**
    - StopLight under kategorin **hue** (detta är en komponent uppbyggd av flera aktörer och behöver installeras för att dyka upp i ActorStore, instruktioner för detta finns i README filen i exemplet)
    - Terminator under kategorin **flow**, denna aktör kan bytas ut mot en Log eller Print aktör för att verifiera status ifrån Lock komponenten.
- Initiera aktörerna som tidigare och koppla ihop dom enligt nedan



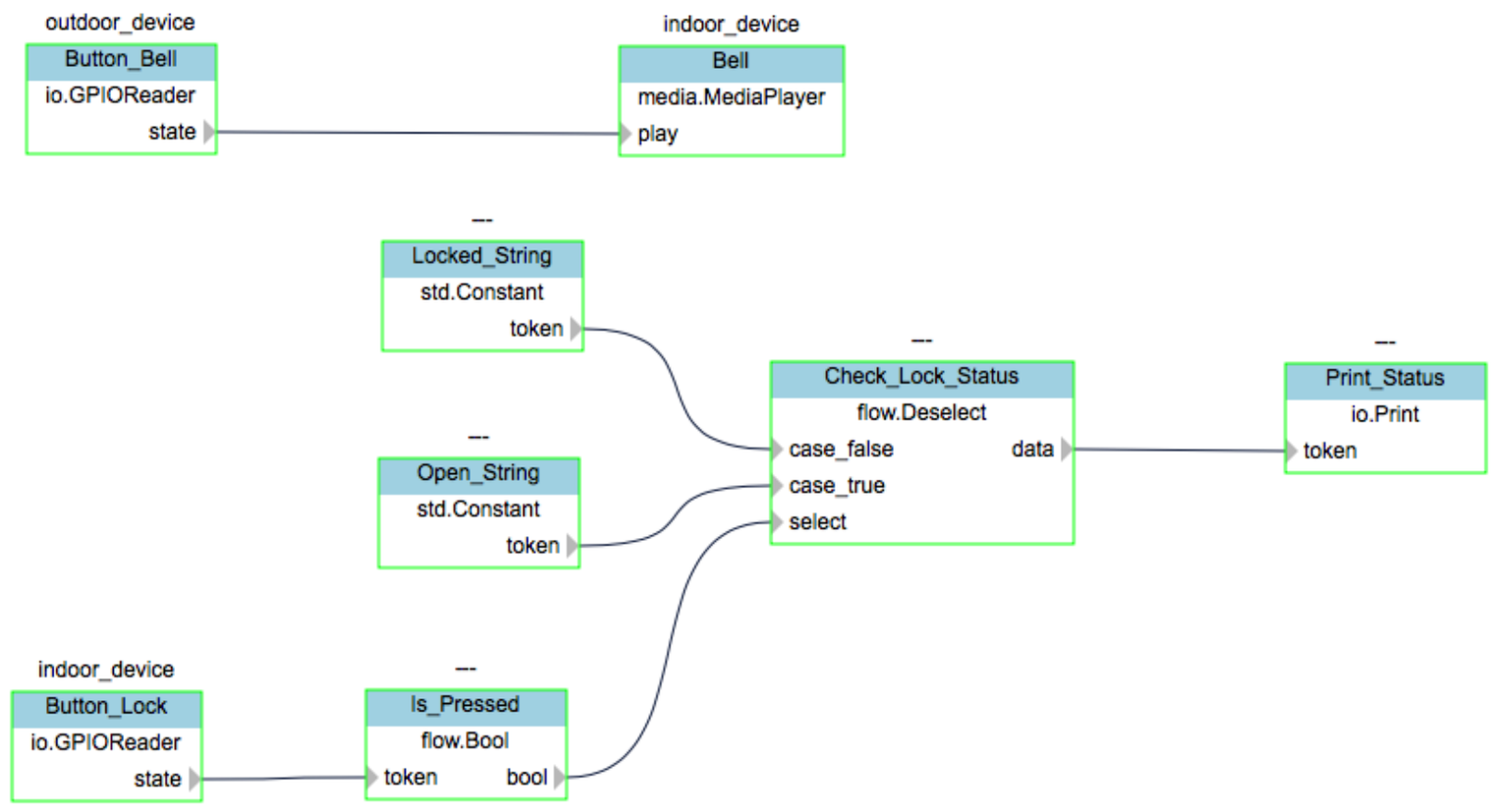
- Starta applikationen och migrera GPIOReader och MediaPlayer aktörerna enligt bilden:



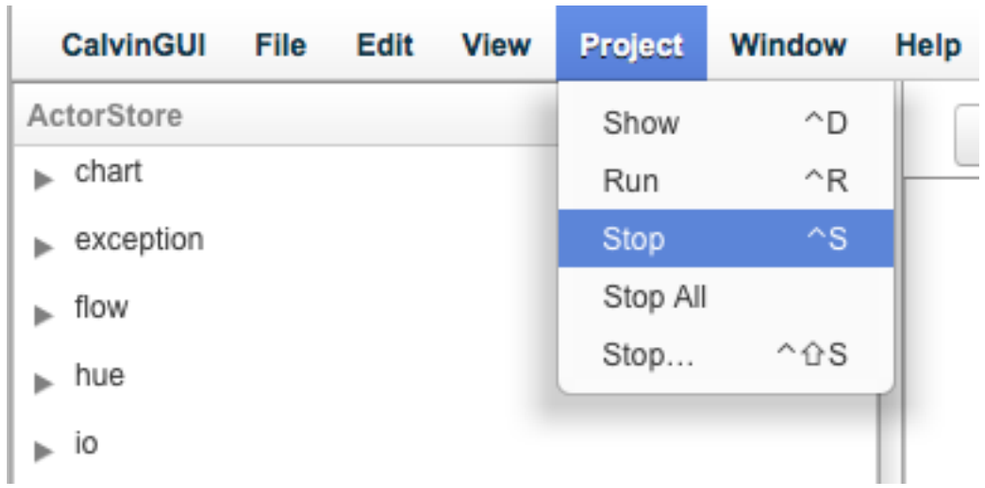
- Testa att trycka på knapparna på de två Raspberry Pierna och se vad som händer.

Lås utan Hue

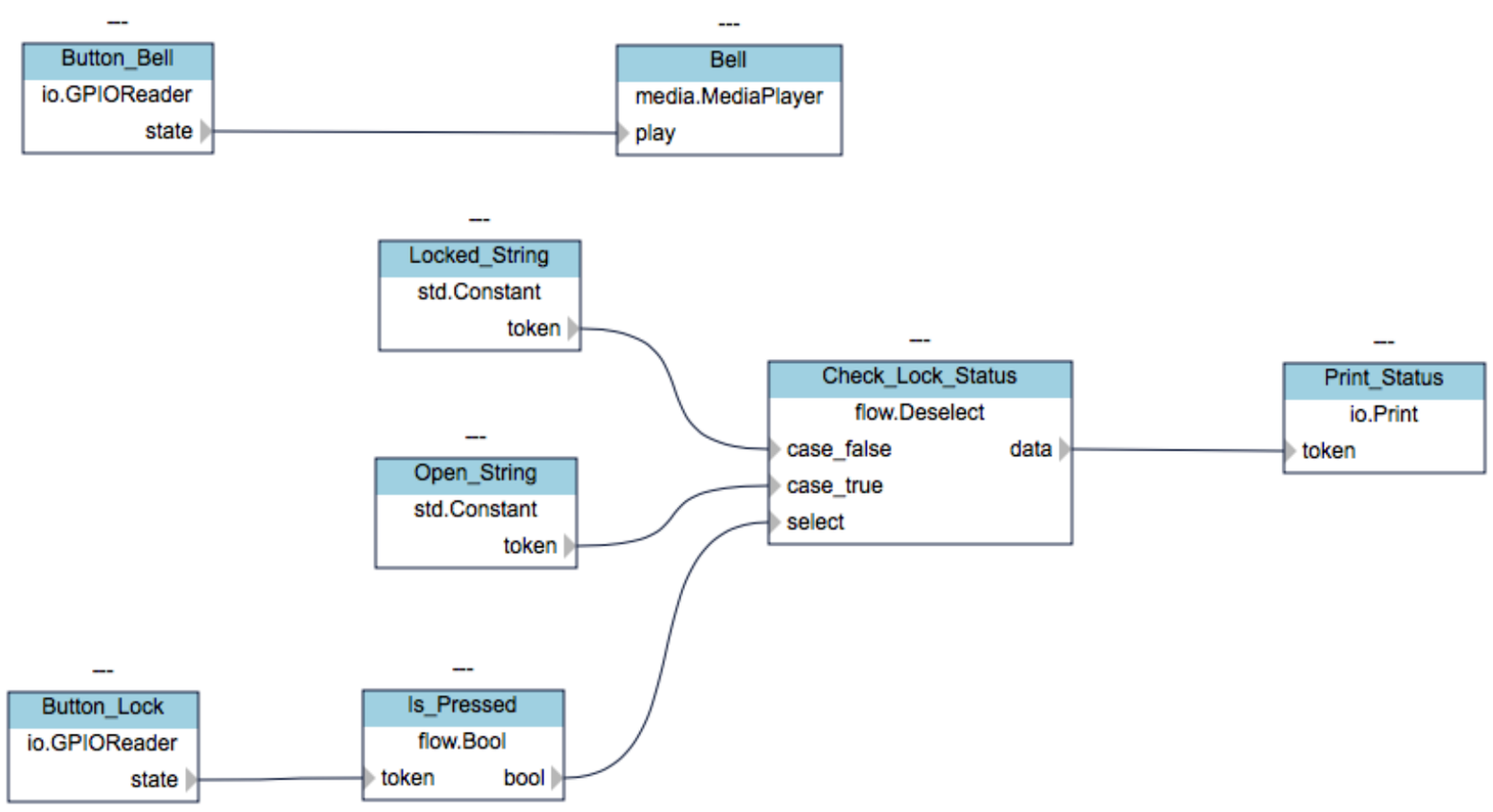
- Applikationen ska nu utökas med några aktörer som tillsammans simulerar ett lås, tillgängligt från insidan via knappen på en Raspberry Pi. En Print aktör kommer skriva status på låset till terminalen.



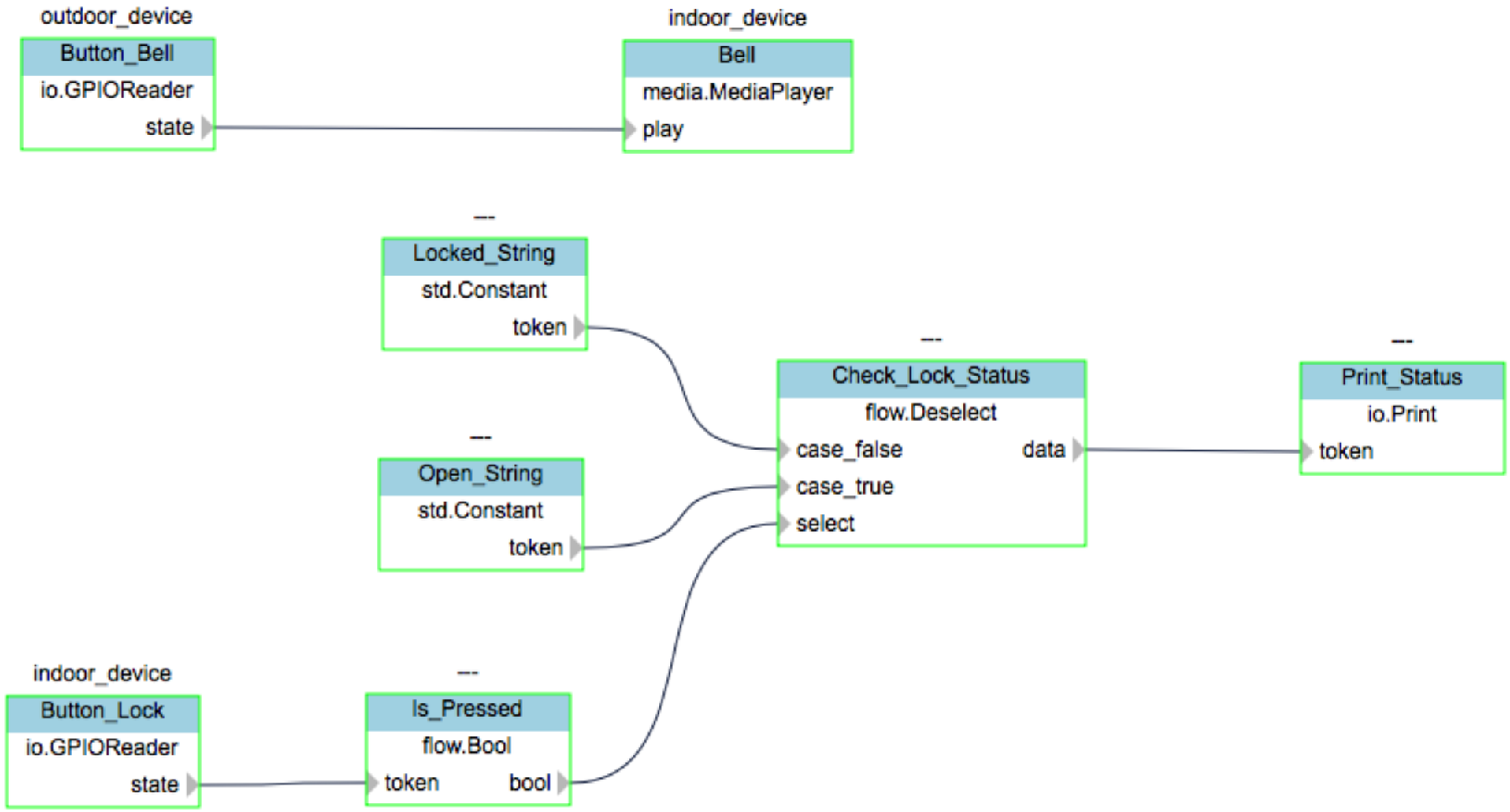
- Start with stopping the last application by choosing **Stop** in the menu **Project**



- Utöver aktörerna GPIOReader och MediaPlayer, dra ut följande nya aktörer till arbetsfönstret:
    - Från kategorin **io:**
        - En Print och en GPIOReader aktör.
    - Från kategorin **flow**:
        - En Bool och en Deselect aktör.
    - Från kategorin **std**
        - Två Constant aktörer.
- Initiera aktörerna som tidigare och koppla ihop dom enligt bilden nedan.



- Starta applikationen och migrera GPIOReader och MediaPlayer aktörerna enligt bilden nedan:



- Testa att trycka på knapparna på de två Raspberry Pierna och se vad som händer.

Om något inte fungerar

- Börja med att verifiera att scripten (part_1 och part_2) ger förväntade resultat. Om de inte fungerar som förväntat, gå igenom README filen och verifiera att alla runtime är korrekt startade.

- Ett enkelt sätta att felsöka en applikation är att lägga in en Identity aktör mellan två aktörer i applikationen. Argumentet **dump** behöver initieras till **true**. Alla tokens kommer då att loggas i terminalen innan de skickas vidare till nästa aktör.