Democracy DEV Board User Doc

Summary

Connecting to Democracy DEV board

Connect the power supply (the SD card need to be inserted in the Raspberry Pi 3). The Board will be running when the green led is blinking.

Then, you can connect to the board through SSH at the IP address: 169.254.0.2 .

Your computer needs a static IP address like: 169.254.0.1 (Subnet mask: 255.255.255.0).

The login is:

- Username: pi
- Password: raspberry

Root User

All the applications and commands are run as root.

To log in root, use the command:

 # sudo su

Shutdown the board

To shutdown correctly the board, please enter the following command (as root):

 # poweroff

After entering this command, the blinking green led will switch off. You can now unplug the power supply.

Audio Relay Bypass

By default, the audio Bypass is enabled, hence the audio from Guitar in is connected directly to Amp Out (Analog bypass) without being processed by the Democracy DEV board.

To disable the audio bypass, a specific GPIO needs to be set. This can be done by a script available in the "/home/pi/morpheus/scripts" directory. You can use the following commands:

 # cd morpheus/scripts

 # ./set-passthrough.sh 0 (You will hear the relay switching)

To enable the audio bypass, please use the following command:

 # ./set-passthrough.sh 1

Load Alsamixer configuration

The alsamixer configuration with all the parameters configured correctly to run the applications is available in the "/home/pi/morpheus/cirrus-scripts" directory.

To load all parameters of alsamixer, enter the command below:

 # alsactl --file asound_morpheus.state restore

After entering this command, all parameters are updated. If you want to modify or check the value of a parameter manually, enter the command below:

 # alsamixer

This will open the menu of alsamixer.

Expression pedal in and out

First, we need to configure the mode used by the expression pedal. This can be done by the script *adg2188-set-mode.sh* available in "/home/pi/morpheus/scripts" directory.

The script *adg2188-set-mode.sh* has two options:

mode:

- 1: Passive TRS (tip expression)
- 2: Passive TRS (ring expression)
- 3: Passive TS
- 4: Active

digipot:

- 10: Use the 10k potentiometer
- 50: Use the 50k potentiometer

As an example to test (mode passive TRS (tip), 10k for digipot out) use the command below:

 # cd morpheus/scripts

 # ./adg2188-set-mode.sh 1 10

A simple application is available in "/home/pi/morpheus/pedal_in" directory. This application reads the value detected on Pedal in, and writes the corresponding value on Pedal out.

To run this application, please use the command below:

 # cd morpheus/pedal_in

 # ./pedal_meter_raw

Pure data application

Two example PureData applications are available in the */home/pi/morpheus/patches* directory of the Raspberry SD card.

In order to modify or run the applications on your computer, you can download the Pure Data software (Pure Data 0.47-1) from the following link: https://puredata.info/downloads/pure-data

The application (patch_receiver_mono.pd) developed for our test allows to switch between 2 audio sources, "MIDI in" (value=2) and "Guitar in" (Value=1) and to connect the audio selected to the output (Amp Out).

If "MIDI in" is selected, the source is a MIDI file played on the raspberry (or it could be the real MIDI from the interface), and the MIDI port is connected to fluidsynth synthesizer to generate a sound at the output.

If "Guitar in" is selected, the sound from Guitar in is directly connected to the output (Amp Out).

To control this application, a second application (patch_sender.pd) needs to run on a computer to send the command selection (1 or 2).

To run a Pure Data application on the Raspberry, please use the script available in "/home/pi/morpheus/scripts" directory through the commands below:

 # cd morpheus/scripts

 # /patches_selector

When the script is running, enter the option *"x",* which will display all the Pure Data patches available in *the "morpheus/patches"* directory.

Choose the patch number 1 (patch_receiver_mono.pd) by entering the value *"1".*

Then, the application will start and run and the message *"Application Running!!!"* will display

After this, you can start Pure Data software (previously downloaded and installed).

Open the application from the menu file->open and select "patch_sender.pd" into the right directory.

When the application is open, you can click on the box "connect 169.254.0.2 52522". This way, the application running on your computer will be connected to the one running on the raspberry.

To control the selected input, click (and keep it pushed) on the box with the value 0 (by default) and move the mouse up or down to change the sent value (be careful, it is very sensitive).

The value (1: Guitar in; 2: MIDI in) is automatically sent. (Don't forget to disable the audio relay Bypass)

To run a MIDI file on the raspberry, enter option *"m"* from the "patches_selector" script.

Then, enter the path of a MIDI file, for example: */home/pi/morpheus/test.mid*

To quit "patches_selector", enter the option "q".

Instrument selector

The application Instrument selector allows the user to change run time the instrument (soundfont) used in Fluidsynth. The available instruments on the board are stored in the folder: "/home/pi/morpheus/instruments".

The soundfont files need to be .sf2 format. Be careful, the file name should not contain space. You can add some other soundfont by copying the file in the directory.

Before starting the application, Fluidsnth must be already running.

To start the application, use the command below:

 # instrument_selector

A menu will be displayed, asking to select an option. To choose a new instrument, enter the option "x".

It will display all the available instruments with an associated number. Write the number of the instrument you want to select and push ENTER.

The instrument is now played by Fuidsynth.

The application asks you again to select an option. You can choose again a new instrument or another option.

To quit "instrument_selector", enter the option "q".

MIDI in with Fluidsynth

The application runs Fluidsynth synthesizer with MIDI interface at the input. MIDI notes can be provided with a MIDI keyboard (or a PC with a MIDI interface).

A software for a virtual MIDI Keyboard is available here: http://vmpk.sourceforge.net

The sound is generated by Fluidsynth.

The "startup_fluidsynth_midi.sh" script for this application is available in the morpheus/tests/midi_in directory. The script runs ttymidi to have alsa midi port available. The script runs also the jackd server and fluidsynth. Also, the Audio bypass relay is automatically disabled at startup.

If jackd server and Fluidsynh are already running, you can stop them with the command below (as root):

 # ./startup_fluidsynth_midi.sh --stop

This command is also used to stop the application. You can execute the script with the command below (as root):

 # ./startup_fluidsynth_midi.sh --start

You can now send MIDI notes on the MIDI interface. You can also use instrument selector to change the instrument.

Send SPI commands manually

To send an SPI command manually to the FPGA, a script is available in "/home/pi/morpheus/spi-fpga" directory.

The script has two options:

-C command_value : command_value corresponds to the address of the sent data. Available address values are between 0 and 15 (4 bits).

-D data_value : data_value corresponds to the sent data. Available data values are between 0 and 15 (4 bit).

The parameters address available in the FPGA are :

- Command(address) = 0 , FPGA audio output selection:
  - Data=0: audio out connect to audio given by raspberry (default).
  - Data=1: audio out connect to audio given by cirrus (loopback in FPGA).
- Command(address) = 1 , Switch on/off FPGA leds:
  - Data=1: Switch On FPGA led1 (Data=0 led off) (led connect to bit(0)).
  - Data=2: Switch On FPGA led2 (Data=0 led off) (led connect to bit(1)).
- Command(address) = 2 , cmd debug (reserved for test, please do not modify it).
- Command(address) = 3 , Guitar threshold up (lsb) (reserved for test, please do not modify it).
- Command(address) = 4 , Guitar threshold up (msb) (reserved for test, please do not modify it).
- Command(address) = 5 , Guitar threshold down (reserved for test, please do not modify it).
- Command(address) = 6 , Synthesizer threshold up (lsb) (reserved for test, please do not modify it).
- Command(address) = 7 , Synthesizer threshold up (msb) (reserved for test, please do not modify it).
- Command(address) = 8 , Synthesizer threshold down (reserved for test, please do not modify it).

As an example, you can use the commands below to activate loopback in FPGA:

 # cd morpheus/spi-fpga

 # ./spi-fpga –C 0 –D 1

To disable the loopback in FPGA and come back to the default configuration:

 # ./spi-fpga –C 0 –D 0

FPGA project

The FPGA project called "*morpheus_fpga.xise*" is generated with ISE 14.7 and is available in this USB stick, in */ise14_7/morpheus_fpga* directory.

All the sources are available in */ise14_7/morpheus_fpga/sources* folder.

The generated "top_mboard_v4.bit" bitstream and the associated file "top_mboard_v4.mcs" to program the flash are available in */ise14_7/flash_fpga* directory.

The procedure to program the flash with an mcs file is:

- Connect a Xilinx programmer to the JTAG connector.
- Open Impact software.
- Double-click on "Boundary Scan" (top left corner).
- Right-click (Right side window) and initialize chain.
- Assign configuration files? Answer: No.
- Close the windows: Programming Properties.
- You will have two components on the JTAG chain, xc6slx9 (FPGA) and xcf04s (flash). If not, there is a problem with the programmer or a bad connection.
- To program the flash, right-click on xcf04s component, and select "Assign New Configuration File…"
- Choose the mcs file (top_mboard_v4.mcs) and click open.
- Right-click on xcf04s component, and Select "Program".
- You will have a message with Program Succeeded at the end.

Now you can poweroff and poweron the board. To shut down the raspberry use the command "sudo poweroff". The FPGA will start up with the new bitstream stored in the flash.

The procedure to generate a new mcs file from a bitstream is as follows:

- Open Impact software.
- Double-click on "Create PROM File (PROM File Formatter)" (top left corner).
- Step 1: Select "Xilinx Flash/PROM"
- Click on the "->" button to go to next step.
- Step 2: Select PROM Family "Platform Flash" (as Default), Device (bits) "xcf04s [4M]"
- Click on "Add Storage device" button.
- Click on the "->" button to go to next step.
- Write an output file name: top_mboard_v4
- Choose an Output File Location (path).
- Keep as default: File Format "MCS", Add Non-Configuration data Files "No".
- Click on "OK" button.
- Click on "OK" to the Add Device window.
- Select the bitstream and click "Open" Button.
- Add another devices file to revision, Answer "No"
- Click on "OK" to continue.
- On the Right Side of the window, Double-click on "Generate File…"
- You will have a message with Generate Succeeded when the file is generated.