DE1-SoC Remote FPGA Access Setup

Overview

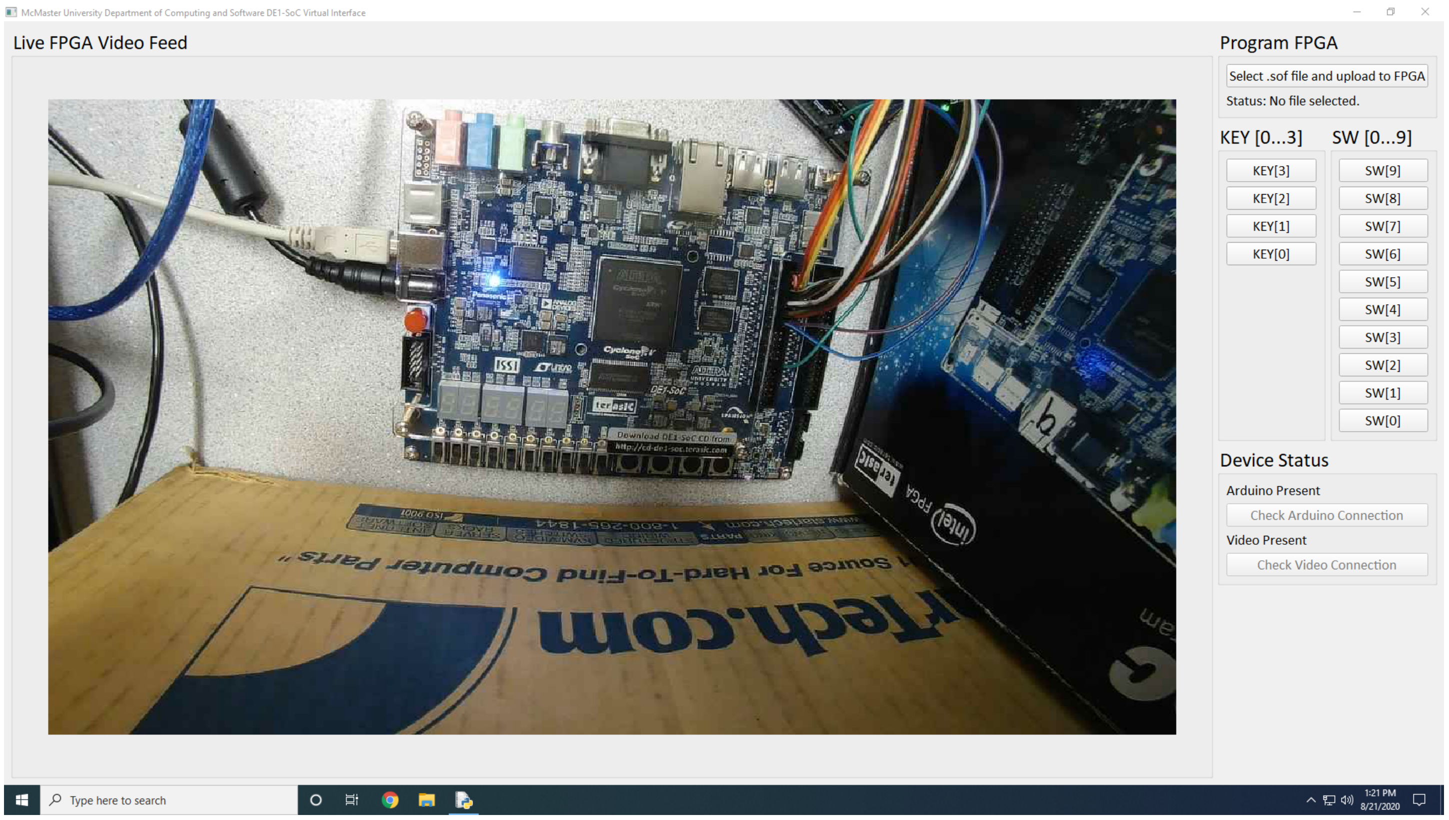

Figure 1: DE1-SoC Virtual Interface

The virtual interface program should look something like figure 1.

The left part of the application, labelled "Live FPGA Video Feed" contains the live video feed from which users can discern the status of the LEDRs and the HEXs.

The top right, labelled "Program FPGA" contains a button called "Select .sof file and upload to FPGA" that opens a file picker for users to choose a .sof file.

The middle right, labelled "KEY [0...3]" and "SW [0...9]" contain toggleable push buttons that emulate the physial switches on the FPGA. These buttons trigger signals on the GPIOs rather than the physical KEYs or SWs; for this reason the remote.qsf pin assignments file must be used rather than the DE1-SoC.qsf file.

The bottom right, labelled "Device Status" contains information about the status of the Arduino and Video feed. If the hardware is working as intended, "Arduino Status" and "Video Status" will be "Working". If the hardware fails, the status will become "Failure" and an error box will appear. The user can attempt to resolve the error by clicking the "Check Arduino Connection" or "Check Video Connection" buttons which only become active after hardware failure. If connection is successful, the buttons will deactivate and an info box will appear. If the connection failure persists, diagnose the errors by viewing stdout - most likely the error is a result of an improper cable connection. If the cables are properly connected and the problem persists, try cycling power to all devices.

Hardware

Required

1. Arduino Uno + USB cable.
2. DE1-SoC Board + Power cable + USB cable.
3. 15 male-female jumpers.
4. USB Camera + USB cable.
5. Computer with 3+ USB ports.

Nice to haves

1. Mounting hardware to fix the camera above the board.
2. Fan to blow air above and below the board if the board will be powered on for long periods of time.
3. Lights around the board/lights on in the room as the camera is better able to discern individual HEX segments in bright light rather than low light.

Arduino - GPIO Jumpers

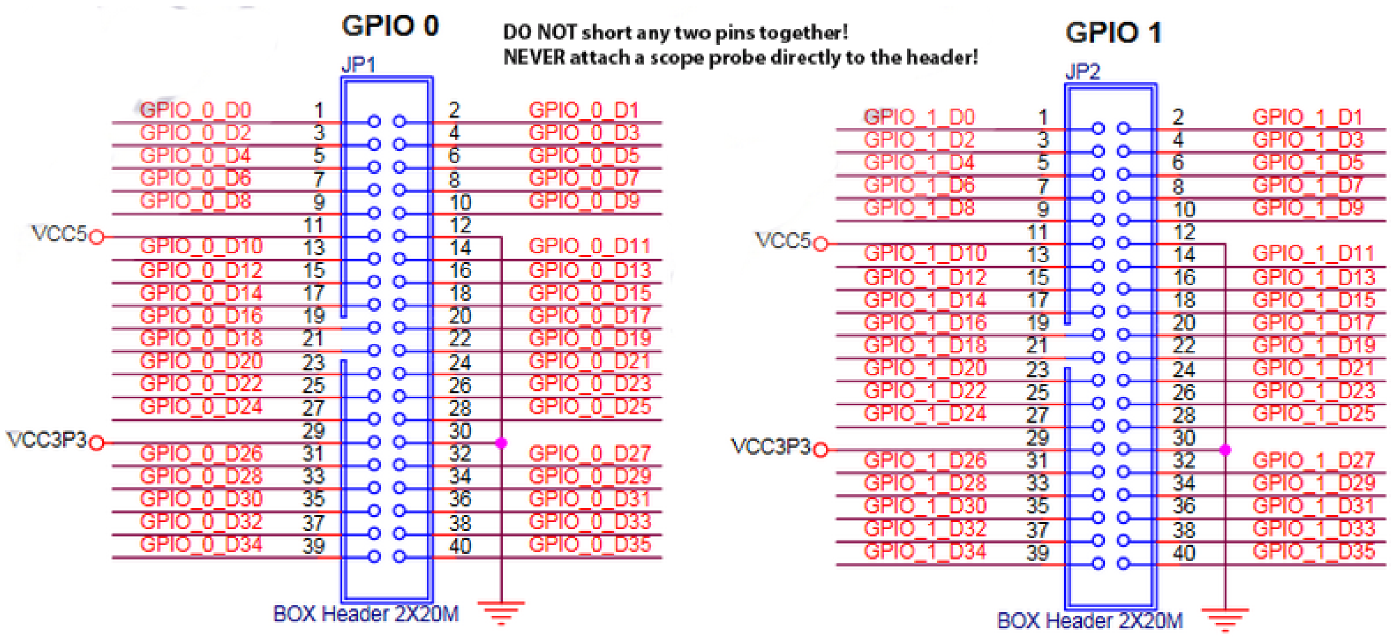

Figure 2: DE1-SoC GPIO Pinout

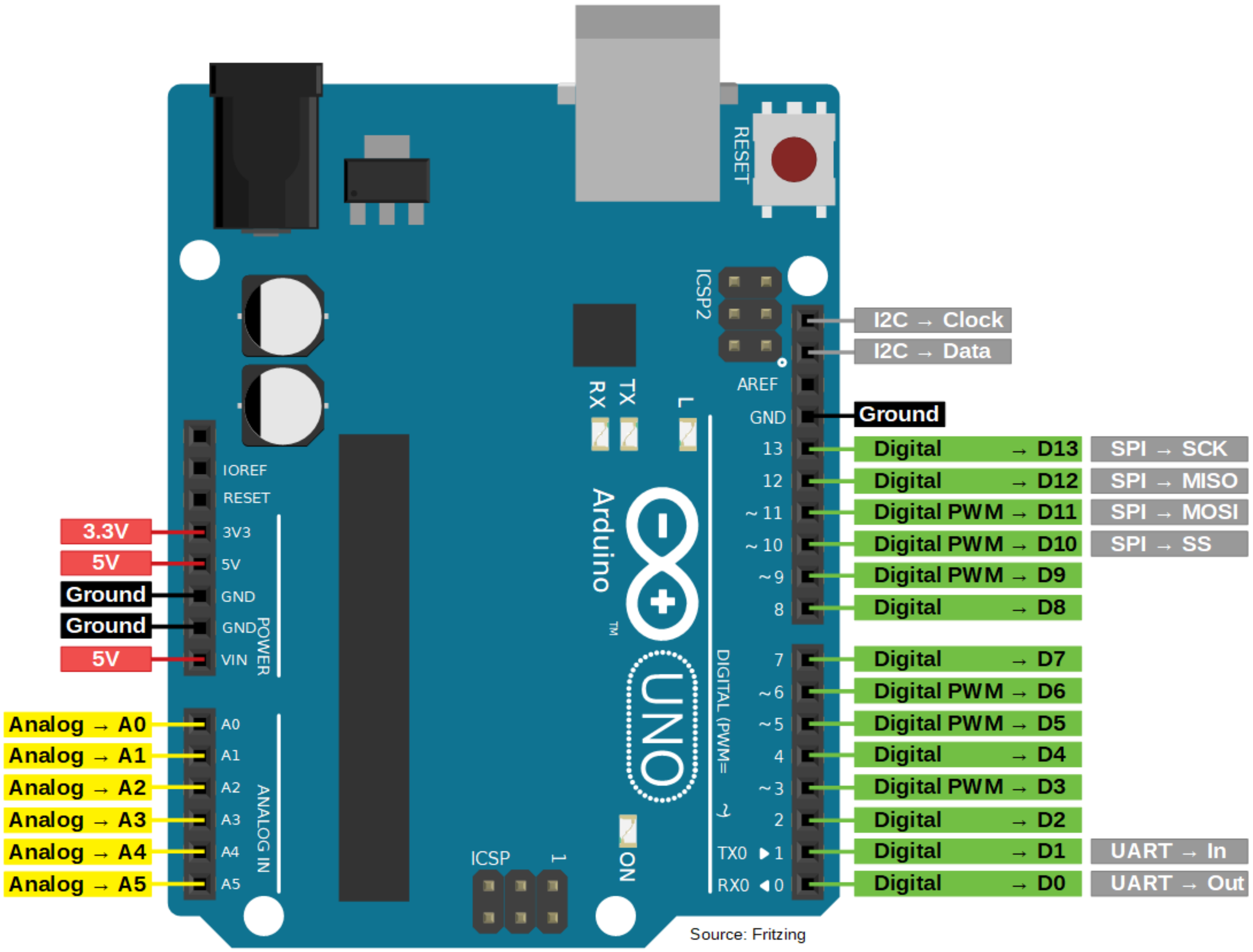

Figure 3: Arduino Pinout

Use GPIO_0. *You may use any GPIOs, but make sure you edit your .qsf file to match. Refer to this GPIO pinout and this Arduino pinout. Make the following connections:

1. Arduino D2 to GPIO_0_D0
2. Arduino D3 to GPIO_0_D1
3. Arduino D4 to GPIO_0_D2
4. Arduino D5 to GPIO_0_D3
5. Arduino D6 to GPIO_0_D4
6. Arduino D7 to GPIO_0_D5
7. Arduino D8 to GPIO_0_D6
8. Arduino D9 to GPIO_0_D7
9. Arduino D10 to GPIO_0_D8
10. Arduino D11 to GPIO_0_D9
11. Arduino D12 to GPIO_0_D10
12. Arduino D13 to GPIO_0_D11
13. Arduino A0 to GPIO_0_D12
14. Arduino A1 to GPIO_0_D13
15. One of the two Arduino GNDs to one of the two GPIO_0 GNDs (pin 12 or pin 30).

Software

Arduino

1. Get Arduino from https://www.arduino.cc/en/main/software.
2. Get v1.ino from https://github.com/JohnsonQu1999/VI_A in the v1 directory and save it to your computer.
3. Connect the Arduino to your computer.
4. Run Arduino, open v1.ino, select the "Tools" menu bar, select "Arduino/Genuino Uno" for "Board", and select the correct port for "Port".
5. Press the upload button - it's an arrow pointing to the right.

Quartus

Only the Quartus Programmer is required for this application to run.

1. Get "Quartus Programmer and Tools" under the "Stand-Alone Software" group in the "Additional Software" tab of this page https://fpgasoftware.intel.com/17.1/?edition=standard. We are using 17.1 but you may wish to get a different version.

2. Install it.

3. Add quartus_pgm to PATH.

   (a) Press Win+S and search for "sysdm.cpl". Press enter.

   (b) Navigate to the "Advanced" tab.

   (c) At the bottom right, click "Environment Variables".

   (d) Under "User variables for USERNAME", click "Path" and click "Edit".

   (e) Click "New" and enter PROGRAMMER_ROOT_DIR\bin64. If you have Quartus Prime installed, enter QUARTUS_ROOT_DIR\bin64.

4. When creating designs, make sure to use the remote.qsf pin assignments file. If you wish to recreate it, start with the DE1-SoC.qsf file and replace KEY[0...3] location assignments with those of GPIO_0[0...3]. Replace SW[0...9] location assignments with those of GPIO_0[4...13].

GUI

1. Get Python from https://www.python.org/downloads/ and install pip when presented with the option.

2. Install the following packages from commandline:

   (a) pip install pySerial (import serial).

   (b) pip install PyQt5.

   (c) pip install opencv-python (import cv2).

   (d) pip install numpy.

3. Add Python to PATH if not already added.

   (a) Press Win+S and search for "sysdm.cpl". Press enter.

   (b) Navigate to the "Advanced" tab.

   (c) At the bottom right, click "Environment Variables".

   (d) Under "User variables for USERNAME", click "Path" and click "Edit".

   (e) Click "New" and enter the directory to your Python executable.

4. Get gui.py from https://github.com/JohnsonQu1999/VI_A and save it in the directory you wish to run it from.

5. If you wish to run the file by double clicking it, right click gui.py, click "Properties", click "Change" in the "Opens with" category, and select pythonw.exe.