Pre-installation

Zephyr version: 3.5 or later CMake Version: 3.27.4 OS: Windows 11 23H2 or later Vivado version: 2023.2
Board: STM NUCLEO-F030R8 (or better)

Installation

Hardware:

[WIP]

STM32

[WIP] If you are using an STM NUCLEO-F030R8 one can simply follow the zephyr tutorial but instead of
`samples\basic\blinky` it should direct to the folder containing the cmake and build files for this project.
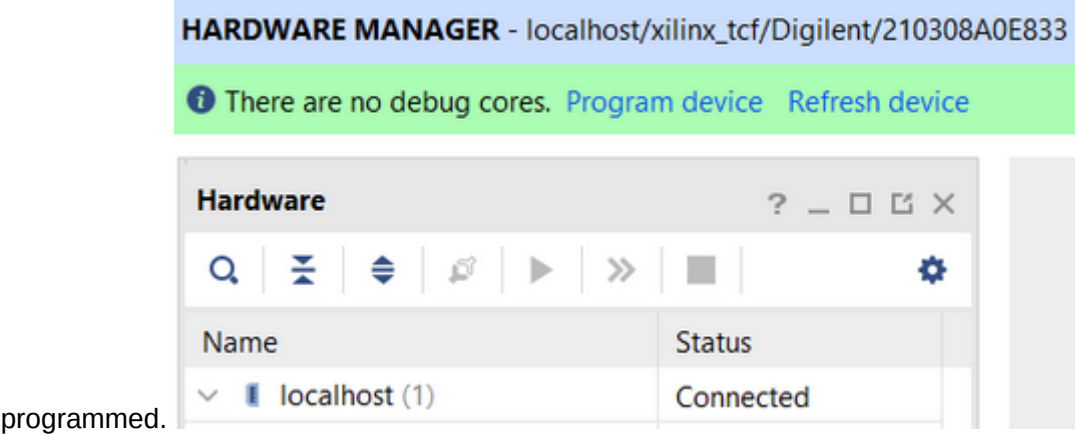
FPGA

To install the game on the FPGA first open the project in Vivado.

1. If there is no bitstream generated generate one. The option to generate a bitscream can be found in
   'PROGRAM AND DEBUG'.
2. A few popups will appear Select OK on everything and wait.
3. Now you will either receive an error message or a popup that the bitstream has been generated. If the
   latter was the case continue to 4. Otherwise, check the error and continue to the topic: errors FPGA
4. In the hardware manager press on 'Open target' if no hardware target is open and press Auto Connect.
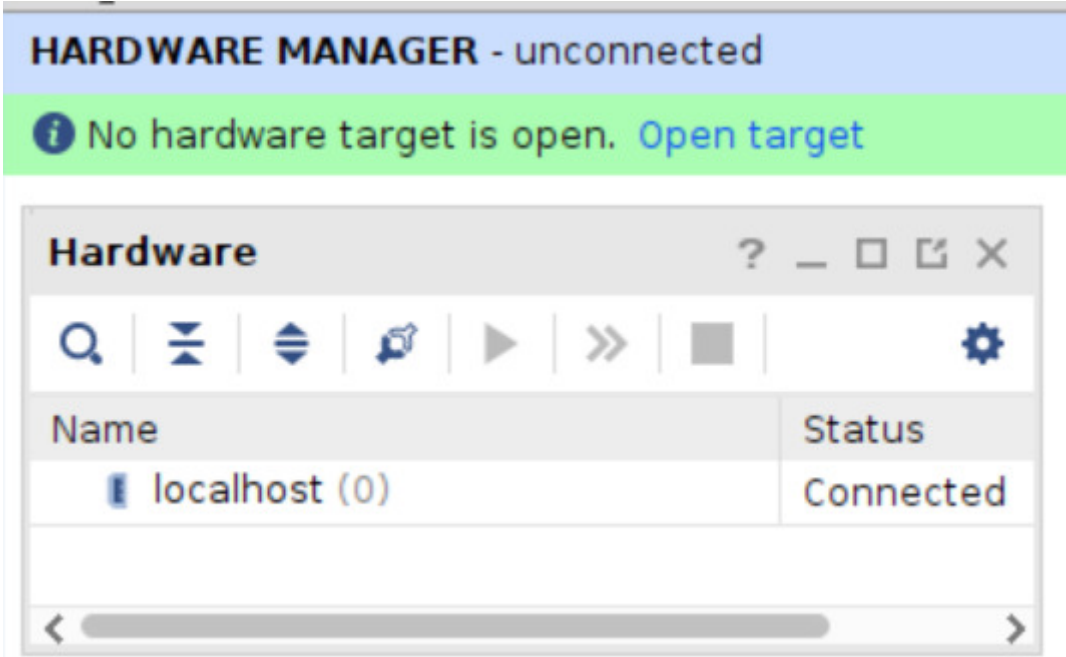
5. After connecting to the FPGA press 'Program device' and then 'OK'. Now the device should get


programmed.

errors FPGA

It is possible that when generating a bitstream results in an error. There are many options however most are the following:

1. "error: no error" This error may be ignored and you can continue as normal.
2. "Constraint error" It is likely that you have accidentally renamed a variable in the main or constraints file or forgotten to add it to the constraints file.
3. "Timing constraints error" You might have added timing constraints to which the device was not built. One should either try to fix these by delving into the code or remove the timing constraints from the constraints file.
4. "Mismatched size ..." Match the size of the mismatched variables. Make sure to modify the correct variable since that has affect on how the program runs and how many variables you might have to change.

Controls

There are 7 buttons for the user to interact with. There are 2 buttons to navigate the shop menu and 4 to navigate around the field. There also is a button to confirm a purchase which places the selected plant from the menu on the location of the selector on the field.
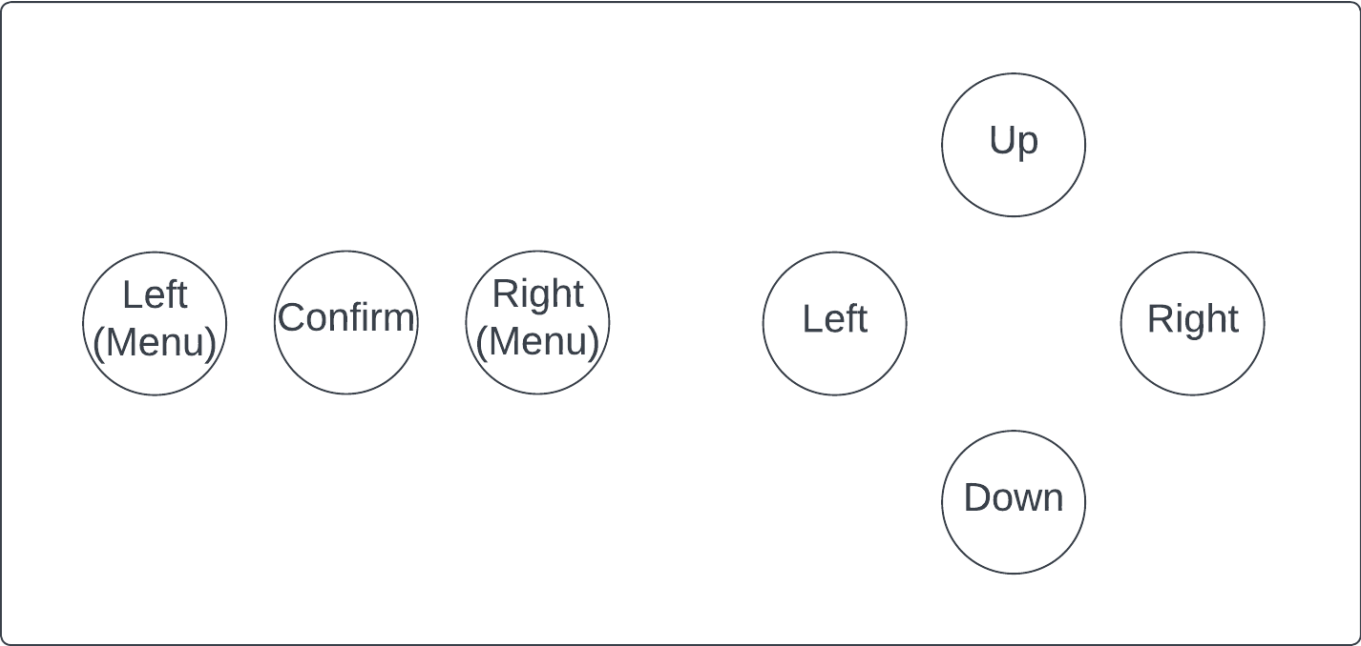
Modifying the program

Update Sprites

Updating the sprites is done by replacing the .coe files in the vivado project. It is important to note that a sprite must be 80x80 pixels

How to convert png's to .coe files

1. locate the file you want to convert
2. download matlab
3. open matlab
4. download the converter (img2coe.m)
5. Replace the *[path/to/file.png]* with the actual path to the file you want to convert.

Replacing the sprite

1. Open the project and locate the sprite you want to replace.

2. Rightclick on the image you want to replace and give the path to the new sprite.
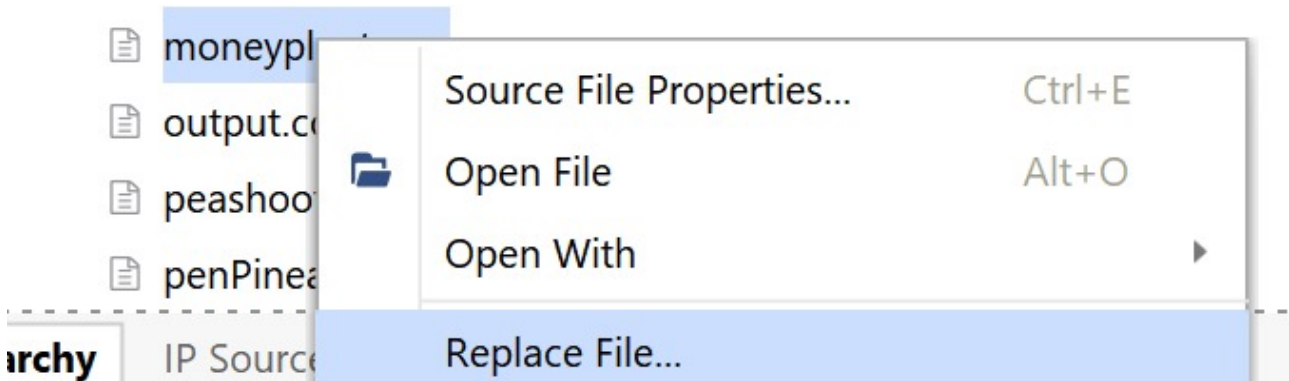
3. In the ROM that matches the sprite you want to replace change the path to the path of your new sprite. These ROM blocks can be found in main->vROM.
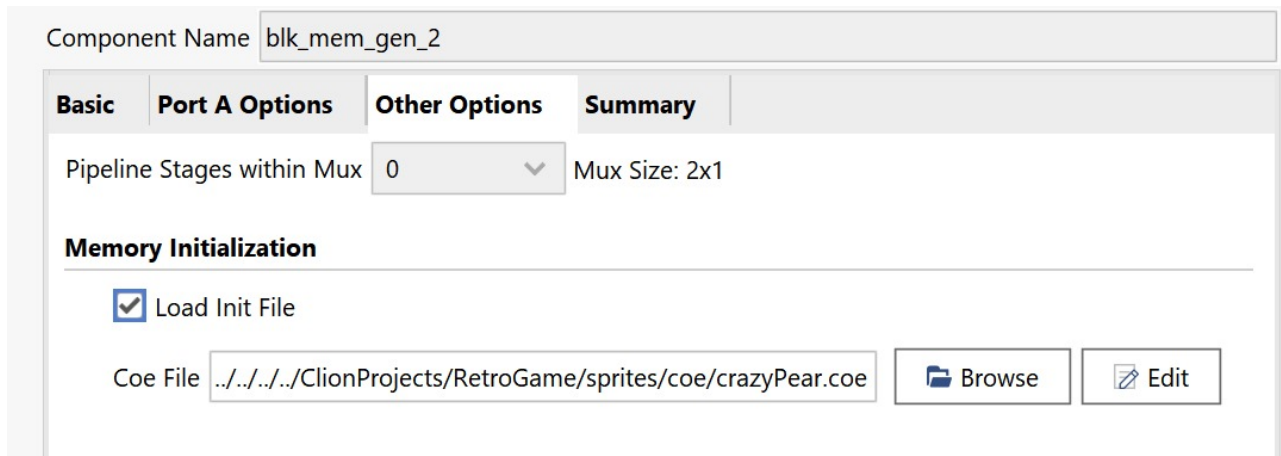
4. After replacing the path let it generate a new Block Memory.

5. When you're sure you've made all changes you can generate a new bitstream and program the device.

Update text

Most text used in the game can be changed from main->textComp. Here one should look for the text they want changed. The text is saved directly in the portmap for simplicity. When changing the text one should be careful and make sure they too change the 'textLength' to the length of the total amount of characters in 'displayText'.

```vhdl
crazyPear : entity work.Pixel_On_Text
    generic map (
        textLength => 3
    )
    port map (
        clk => clk,
        displayText => " 50",
        position => (350, 95), -- text position (top left)
        horzCoord => to_integer(UNSIGNED(h)),
        vertCoord => to_integer(UNSIGNED(v)),
        pixel => displayNr(2) -- result
    );
```

After changing the length of 'displayText' and 'textLength' it might be necessary to change the horizontal position of the text. This can be done in 'position'.

Change music

[WIP]

Change damage sound

[WIP]