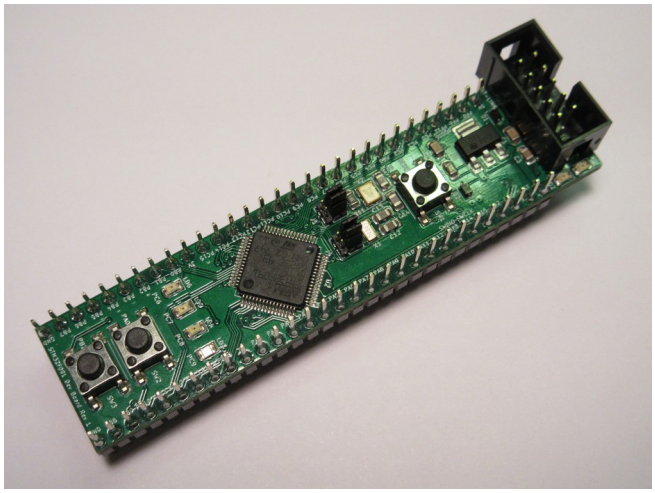
ECE 362 STM32F091 Development Board User Manual

Rev 1.0
August, 2020

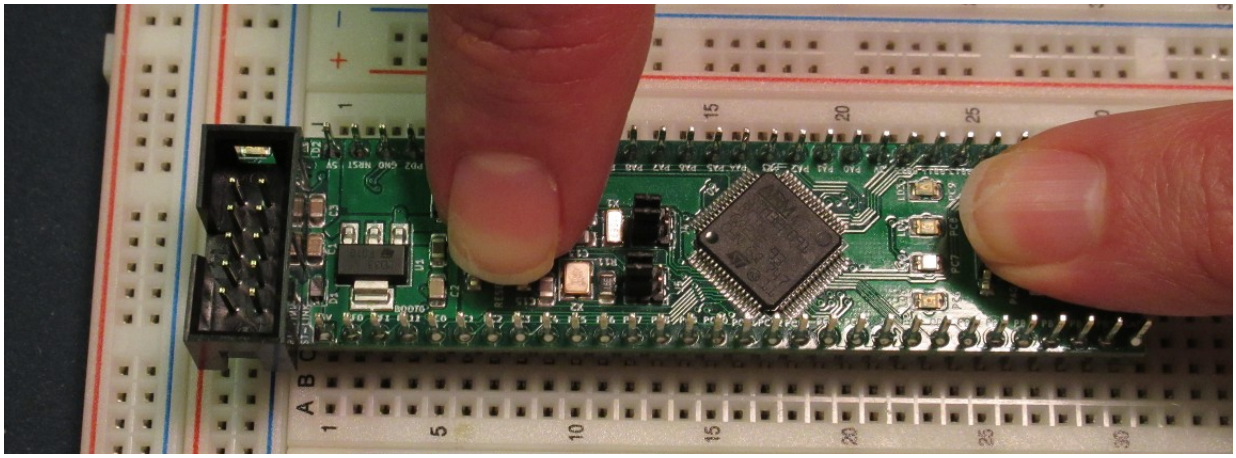
Features



- STM32F091RCT6 microcontroller with 256 KB Flash memory, 32 KB SRAM in an LPQF64 package
- External programmer socket to configure and supply power to the system
- Selectable 8 MHz and 32.768 kHz crystal oscillators or I/Os routed to header pins
- Breadboard-friendly header pins for all I/O pins on the STM32F091RCT6
- External power supply pins for 3 V and 5 V
- Two microcontroller-readable push buttons: SW2 (PA0) and SW3 (PB2)
- Two power indicator LEDs (red: 5 V, green: 3 V)
- Four microcontroller-writable LEDs:
  - LD6 (PC6) Red LED
  - LD7 (PC7) Yellow LED
  - LD8 (PC8) Green LED
  - LD9 (PC9) Blue LED

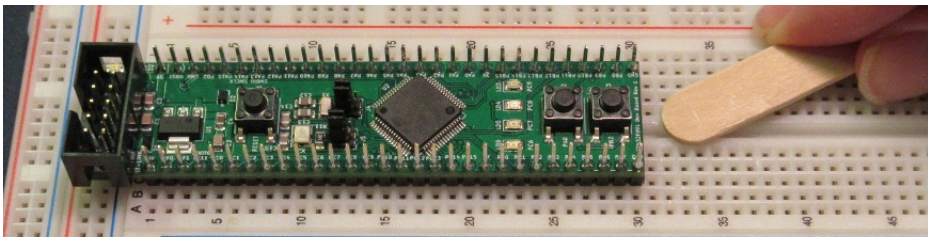
Inserting the Development Board into a Breadboard

When the development board is centered on a typical solderless breadboard with a 0.3” trough, it leaves two holes on either side to connect wires. Use a thick wire to pre-widen the holes where the development board will be plugged in. Place the board on the holes and push it into the breadboard by pressing on the reset and user push buttons. Avoid pressing on the other electronic components.



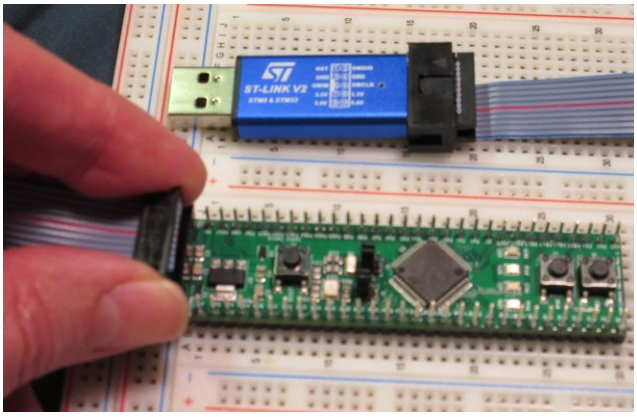
Removing the Development Board from a Breadboard

To remove the board from a breadboard, gently pry one side at a time a short distance with a thin, non-metalic wedge such as a wooden popsicle stick or tongue depressor. Do not pull one side entirely out of the breadboard or the pins on the other side will be bent.

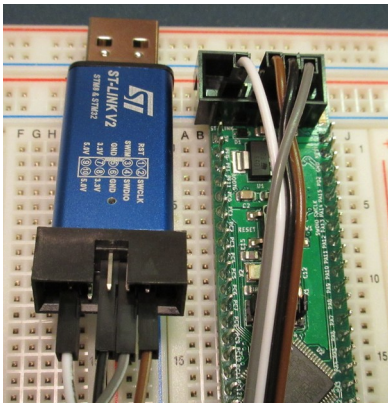


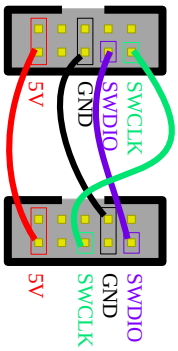
Attaching the Programmer

A 10-pin ribbon cable connects the programming header of the development board to an ST-LINK V2 USB programmer with the proper pin configuration. A 28 gauge ribbon cable will supply enough power for the development board as well as 100 mA of power for external devices connected to the development board. There are two kinds of ST-LINK V2 USB programmers. A programmer with the proper configuration is shown in the image with the check mark. A different programmer pin configuration will neither damage the board nor make it function.



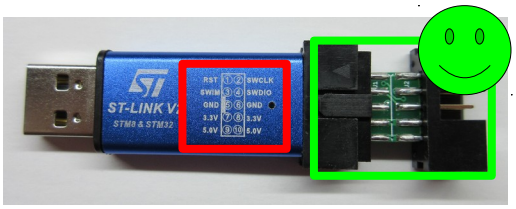
To connect a programmer with the wrong pin configuration, use the four wires that came with the adapter to match each wire in the programmer with the corresponding pin on the development board. Connect a either of the two 5V pins, either of the two GND pins, SWDIO, and SWCLK.



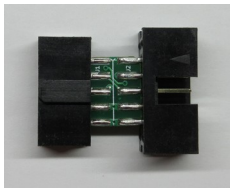


"Wrong" programmer header

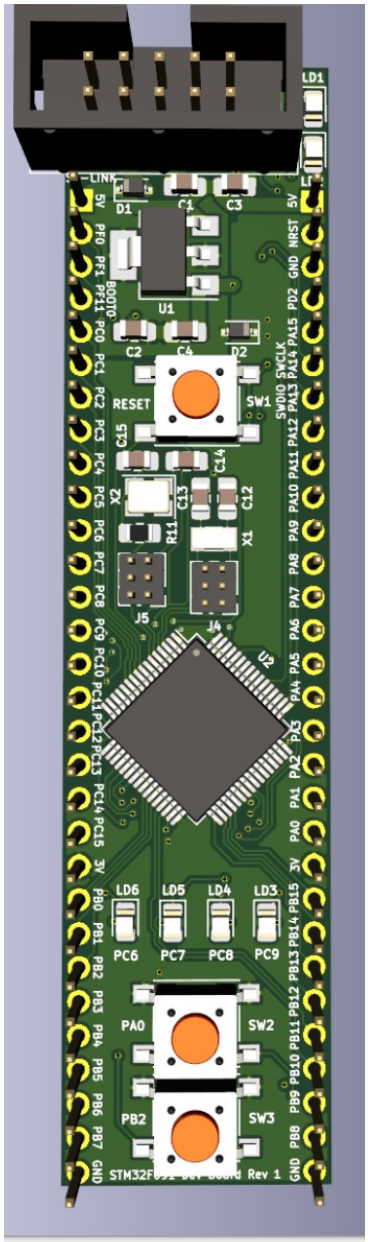
Development board header



Everyone involved with the design, manufacture, and distribution of this development board learned of the existence of a second configuration of programmer a few days before the lab kits shipped to students. We learned this because none of the programmers worked. (Why should there be two different pin configurations of the programmers?) It was too late to do anything that would not greatly delay the delivery. We're building an adapter to correct the pin order. You will receive the adapter as soon as one is built for you.



Using the Development Board



When the development board is properly powered, the red (LD1) 5V indicator and the green (LD2) 3V indicator LEDs near the programming header will illuminate.

Push Button Inputs

The RESET (SW1) button resets the the microcontroller on the development board.

There are two program-readable buttons:

- SW2 is connected to Port A, pin 0 of the microcontroller via a 1K series resistor.
- SW3 is connected to Port B, pin 2 of the microcontroller via a 1K series resistor.

Each push button sets the microcontroller pin high. There is no pull-down resistor to pull the pin low when a button is released. Use the internal pull-down resistors in the microcontroller.

LED Indicators

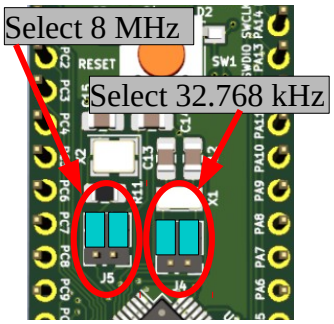
In addition to the power indicator LEDs, LD1 and LD2, four program-writable LEDs are illuminated by a high voltage on the corresponding microcontroller pin.

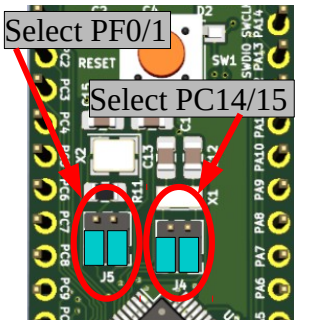
- LD6 is connected to Port C, pin 6.
- LD5 is connected to Port C, pin 7.
- LD4 is connected to Port C, pin 8.
- LD3 is connected to Port C, pin 9.

Less than a milliampere of current is needed to illuminate these LEDs, so they do not interfere with using the pins for general-purpose interfacing.

Crystal Resonator Selection Shunts

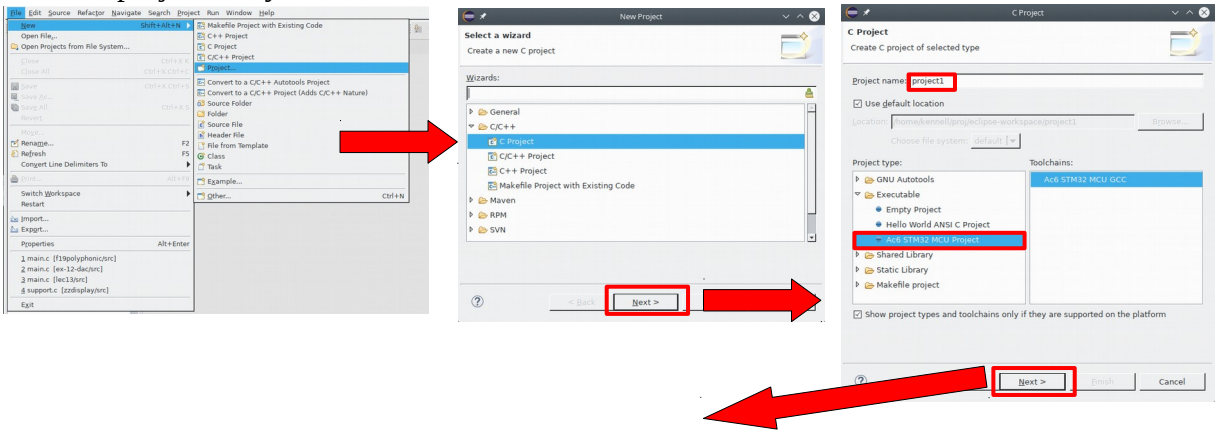
The development board includes two high-precision crystal resonators. An 8 MHz resonator can be connected to microcontroller pins PF0 and PF1 to provide the high-speed external (HSE) clock reference. A 32.768 kHz resonator can be connected to pins PC14 and PC15 to provide the low-speed external (LSE) clock reference. Each of these pins can alternately be used as general-purpose I/O. The function of the pins is selected by inserting shunts into the J4 and J5 6-pin 1.27 mm headers. The development board can run with all jumpers removed so that PC14, PC15, PF0, and PF1 are not connected.

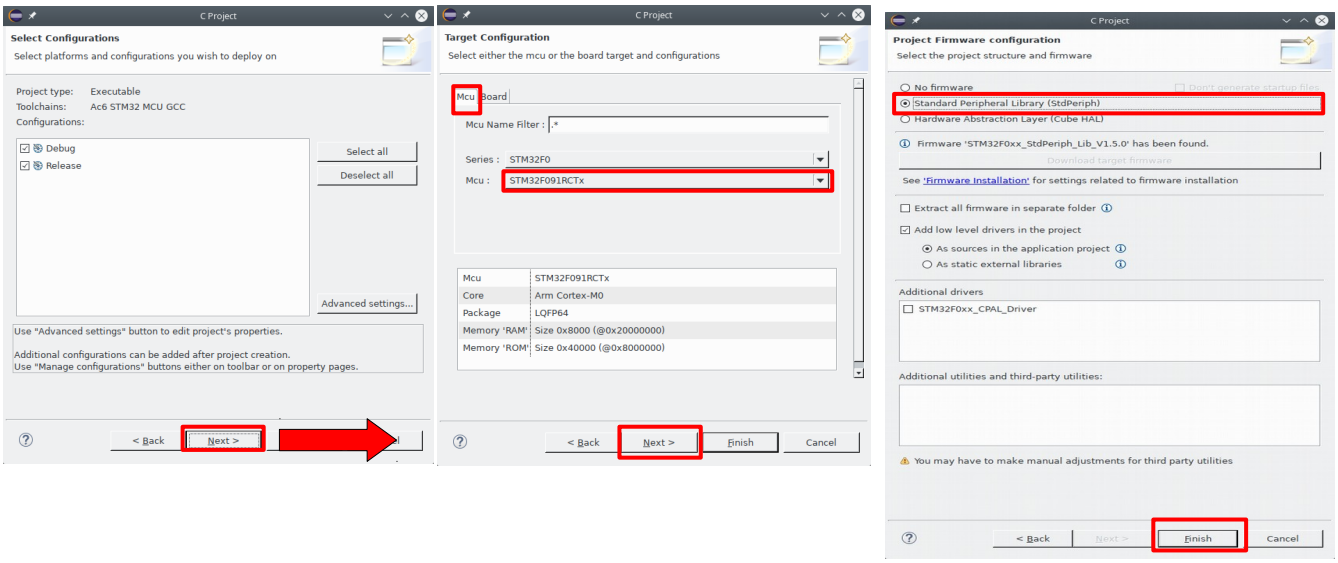




Programming the Development Board

Create a project in SystemWorkbench:





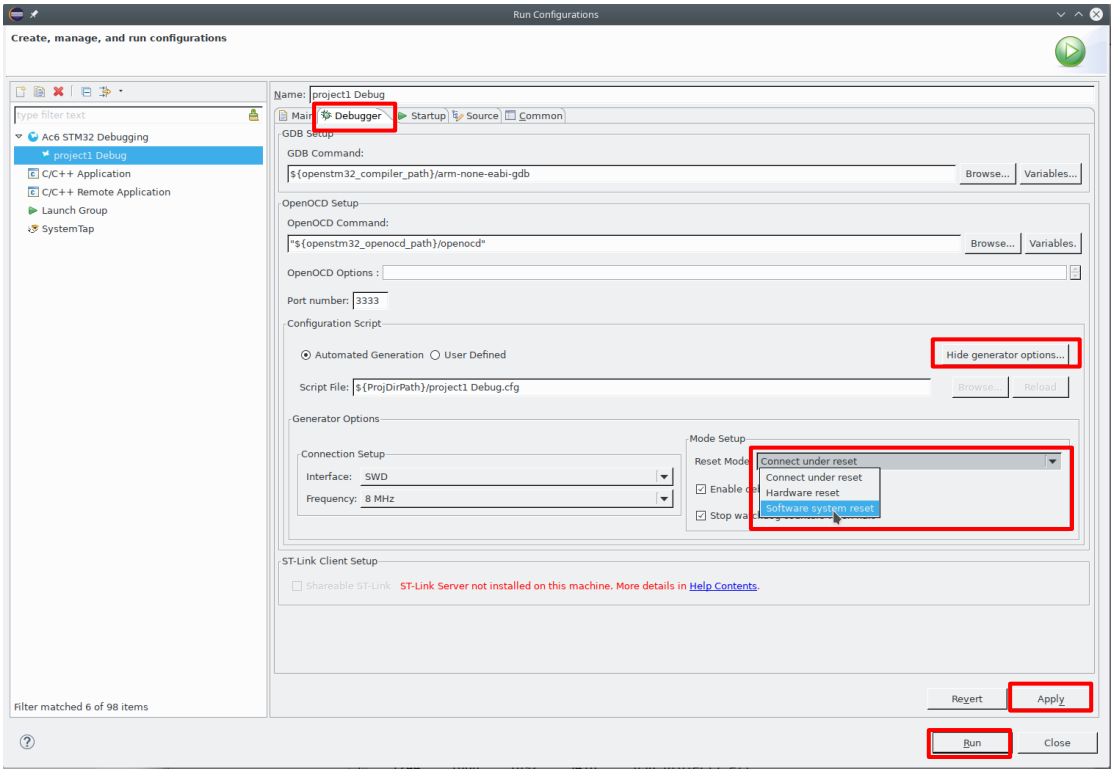
Example Program

```
#include "stm32f0xx.h"

int main(void)
{
    RCC->AHBENR |= RCC_AHBENR_GPIOCEN;
    GPIOC->MODER |= 0x55 << 12;
      for(;;) {
          GPIOC->ODR ^= 0x3c0;
          for(int x=0; x<1000000; x++)
              ;
      }
}
```

Configure the USB ST-LINK V2 Programmer

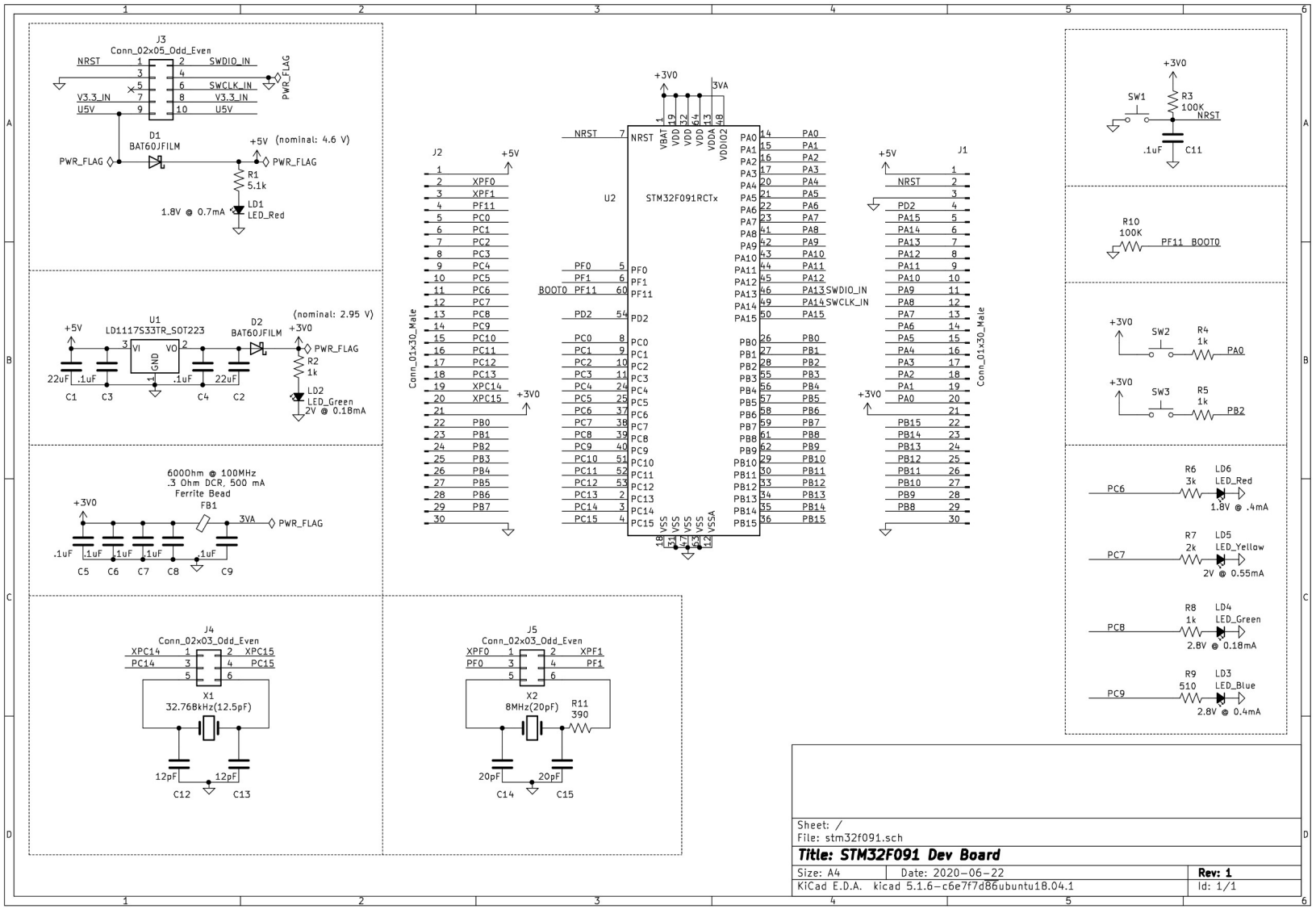
The USB ST-LINK V2 Programmer uses a two-wire SWDIO/SWCLK protocol to interact with the STM32F091 microcontroller. It does not use the reset pin. SystemWorkbench must be configured to use "Software system reset" instead of "Connect under reset", which is the default for ST microcontrollers. To do this, compile your program and select the Run => Run Configurations menu so that the following dialog appears:



Common Problems

Recovering from modification of Port A 13/14 mode: The USB programmer does not use the reset line. Instead, it uses a software system reset that involves a special signal imposed on SWDIO and SWCLK. These two signals are handled with pins PA13 and PA14 of the STM32F091. If the PORTA MODER values for these pins are changed, the microcontroller cannot be programmed, debugged, or reset by the USB programmer. No damage done. To recover, change the program configuration, and hold the Reset button while programming the microcontroller.

Do not use Port C pins 13/14/15 as outputs: Pins PC13, PC14, and PC15 are controlled by a low-power subsystem of the STM32F091. Connections to PC14 and PC15 are selected by shunts in header J4, but PC13 is always connected to a boundary pin on the development board. If any of these pins is configured for output and source or sink more than 3 mA, the driver for the low-speed external (LSE) crystal resonator will be permanently damaged. (See the obscure footnote 1 of Table 13 on page 40 of the STM32F091xBC datasheet.) Thereafter, the pins will still be useful for (low-current) GPIO. Only the oscillator will be damaged. There is no way to repair the oscillator once damaged. It is possible to build an external oscillator that is just as good, but not as convenient.



Sheet: /		
File: stm32f091.sch		
Title: STM32F091 Dev Board		
Size: A4	Date: 2020-06-22	**Rev: 1**
KiCad E.D.A. kicad 5.1.6-c6e7f7d86ubuntu18.04.1		Id: 1/1