Copyright © 2019
PulseRain Technology, LLC.

10555 Scripps Trl, San Diego, CA 92131
858-877-3485 858-408-9550
http://www.pulserain.com

Doc# QSG-0932-0069, Rev 0.9.1

PulseRain Reindeer RISC-V Soft CPU

Quick Start Guide

--Validated with

Efinix Trion T20 BGA256

Development Kit

May, 2019

.

Table of Contents

1 Introduction

1.1 System Overview

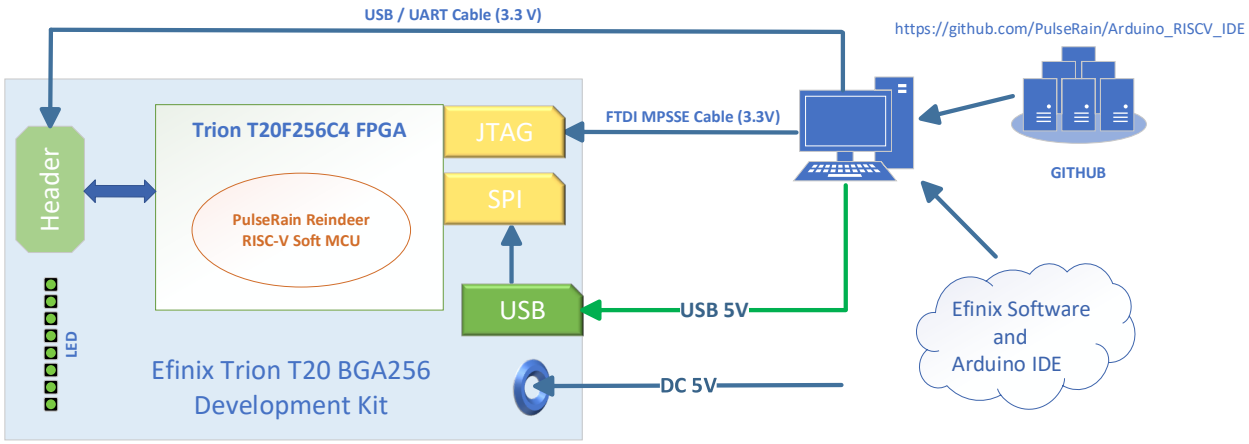

Figure 1-1 System Overview

PulseRain Reindeer is a soft CPU of Von Neumann architecture. It supports RISC-V RV32I[M] instruction set, and features a 2 x 2 pipeline. It strives to make a balance between speed and area, and offers a flexible choice for soft CPU across all FPGA platforms.

And at this point, it has also been successfully ported to the Efinix Trion T20 BGA256 Development Kit ("T20 Dev Kit" for the rest of the document), with the following configurations:

- RV32I processor core, Von Neumann Architecture
- 64KB Block RAM for code and data
- 1 UART TX
- 32 bit GPIO

As shown in Figure 1-1, to further facilitate the software development for RSIC-V, an Arduino board support package has been provided on GitHub for the T20 Dev Kit, with which the software developers can write code for RISC-V directly in Arduino IDE.

And the following components can also be found on the T20 Dev Kit, as illustrated in Figure 1-1.

- A Trion T20F256C4 FPGA
- JTAG and SPI Flash for FPGA configuration
- 8 LEDs
- Connectors for FPGA IO ports
- FTDI 2232H for USB connection

1.2 Supply of Power

The T20 Dev Kit can be powered either through USB cable or through a 5V DC Jack.

1.3 Programming Cable

As indicated in Figure 1-1, a programming cable is needed to download FPGA image to the FPGA through JTAG or SPI Flash. And it is recommended to use the following cable:

C232HM-DDHSL-0, MPSSE Cable (3.3V) from FTDI Chip

1.4 USB / UART Cable

As indicated in Figure 1-1, to let PulseRain Reindeer work with Arduino, a USB/UART cable is needed. It is recommended to use one of the following cables:

- TTL-232R-3V3 Cable from FTDI Chip
- TTL-232R-RPi Debug Cable for Raspberry Pi from FTDI Chip

And the rest of the document will assume TTL-232R-3V3 Cable is used. The pins on the cable should be connected to the H2 connector on the T20 Dev Kit, as illustrated in Figure 1-2.

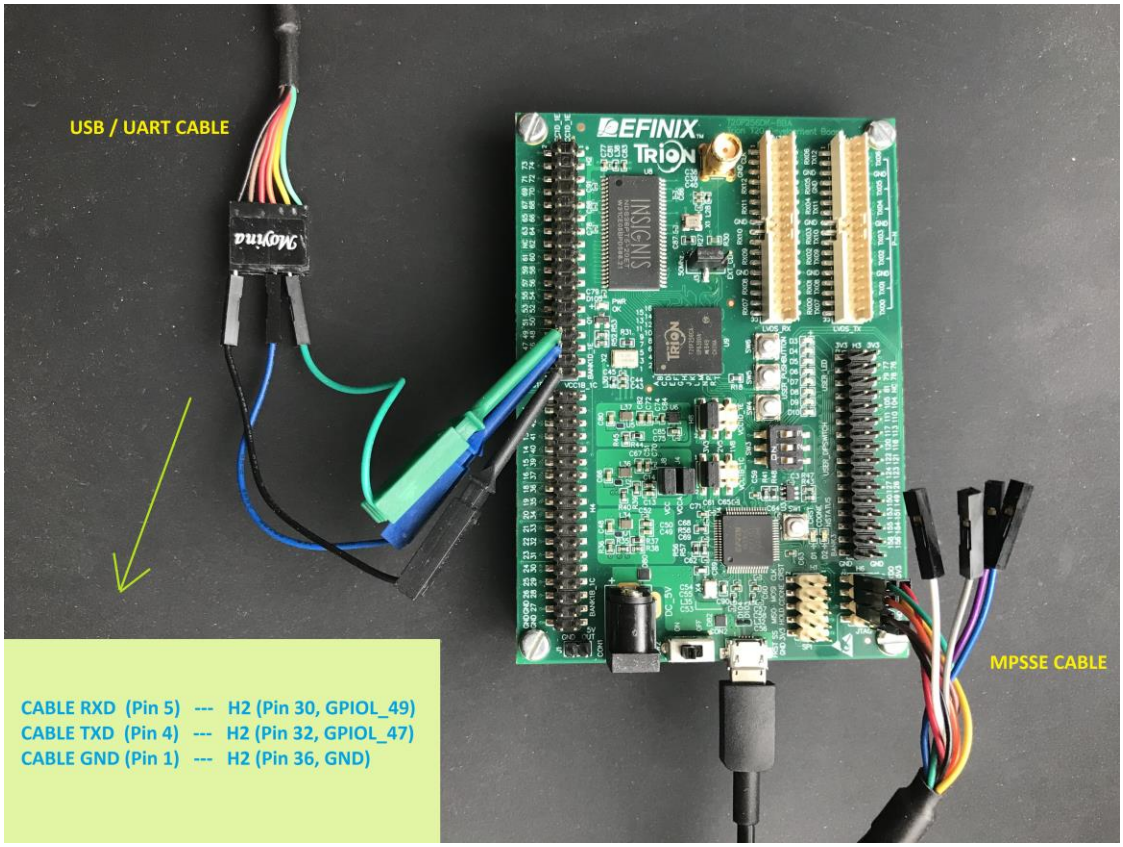

Figure 1-2 Cable Connection

2 Simulate PulseRain Reindeer

The RTL code of PulseRain Reindeer for Trion can be found on GitHub at

https://github.com/PulseRain/Reindeer_Trion

To run the RISC-V compliance test in Modelsim, do the following:

1. git clone https://github.com/PulseRain/Reindeer_Trion.git
2. launch modelsim and enter Reindeer_Trion/sim/modelsim
3. "do build_soc.do" to build the RTL code
4. "do run_compliance.do" to run the compliance test

3 Build PulseRain Reindeer

To synthesize, place/route and generate bitstream for PulseRain Reindeer, do the following:

1. git clone https://github.com/PulseRain/Reindeer_Trion.git
2. launch Efinity software. The rest of the document assumes Efinity 2019.1 is used.
3. Open the project: Reineer_Trion/build/Reindeer_Trion.xml
4. Open Interface Designer as shown in Figure 3-1.

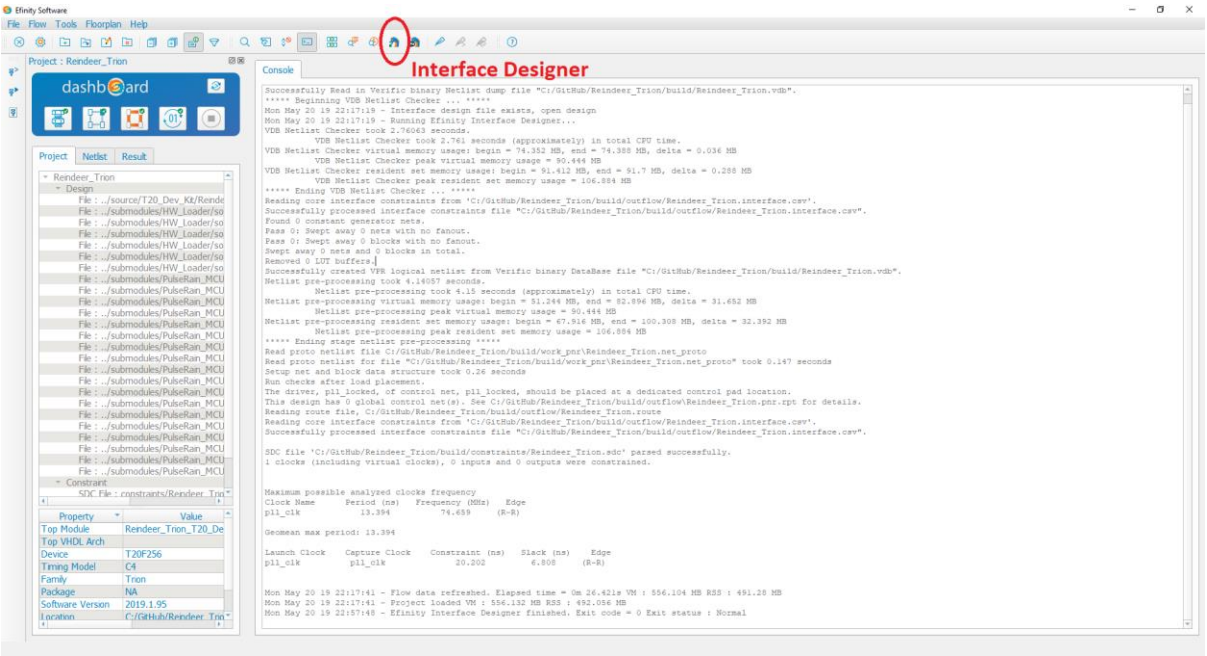

Figure 3-1 Launch Interface Designer

5. Generate Efinity Constraint Files, as indicated in Figure 3-2.

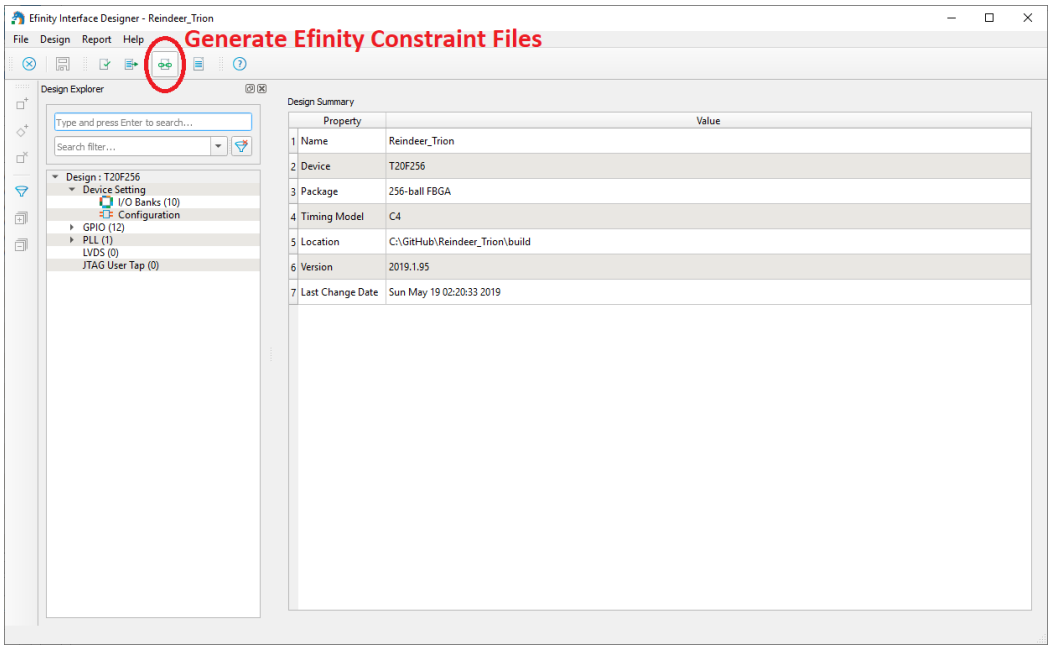

Figure 3-2 Generate Efinity Constraint Files

6. Close the Interface Designer.
7. Start synthesis, Placement, Routing and Generate bitstream
8. The final bitstream can be found in Reineer_Trion/build/outflow/Reindeer_Trion.hex

4 Program the T20 Dev Kit through JTAG

To program the T20 Dev Kit with the bitstream built early, do the following:

1. Connect FTDI MPSSE Cable to the T20 Dev Kit's JTAG header (Figure 1-2) and the PC
2. Go to https://zadig.akeo.ie/ and download Zadig tool
3. Launch Zadig, click menu/Options/List All Devices, and choose C232HM-DDHSL-0
4. Select libusbK and replace Driver

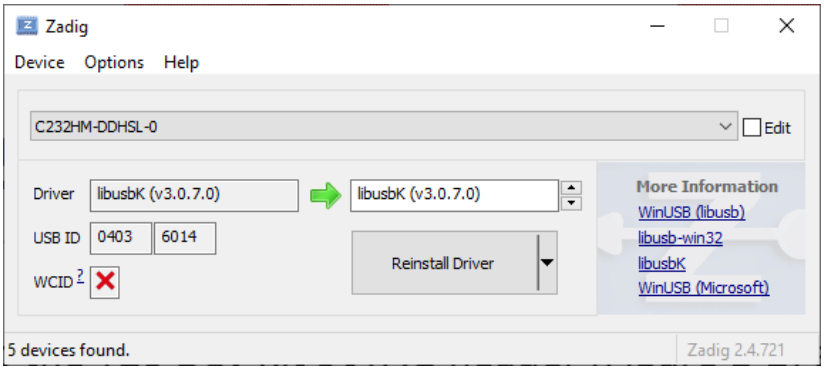

Figure 4-1 Use Zadig to install Driver for MPSSE Cable

5. Set Power Switch S2 to the “off” position, and then to the “on” position
6. Launch Programmer in Efinity Software, as shown in Figure 4-2.

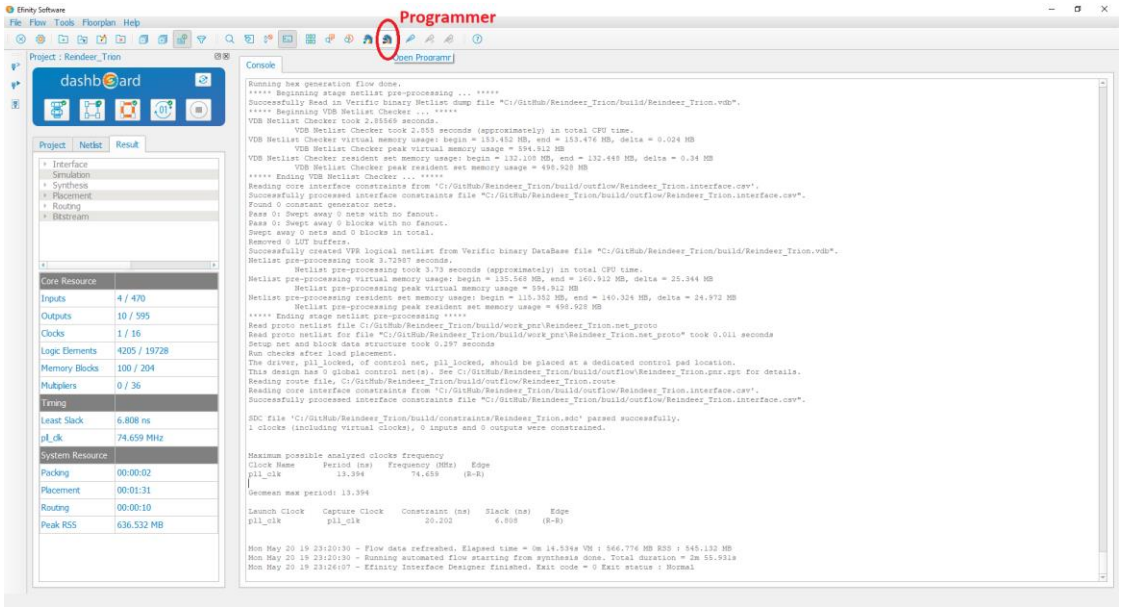

Figure 4-2 Launch Programmer

7. Set the programmer as shown in Figure 4-3. (The image path might be different on each PC)

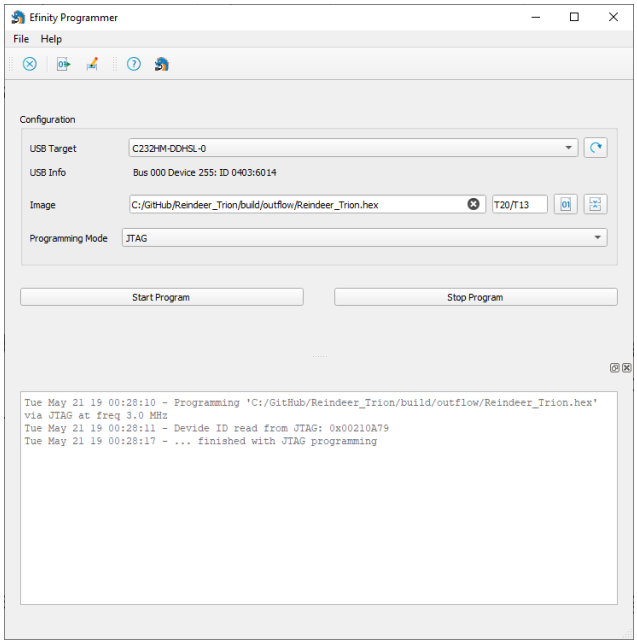

Figure 4-3 Programmer Setup for JTAG

8. Click “Start Program” to program the T20 Dev Kit.

5 Using Arduino IDE

5.1 Connect the Board with USB/UART Cable

As indicated early Figure 1-2, a USB/UART cable is needed to connect the T20 Dev Kit to PC.

The first time an USB/UART cable is connected to PC, Windows may prompt a message dialog for driver installation, like the one shown in Figure 5-1.

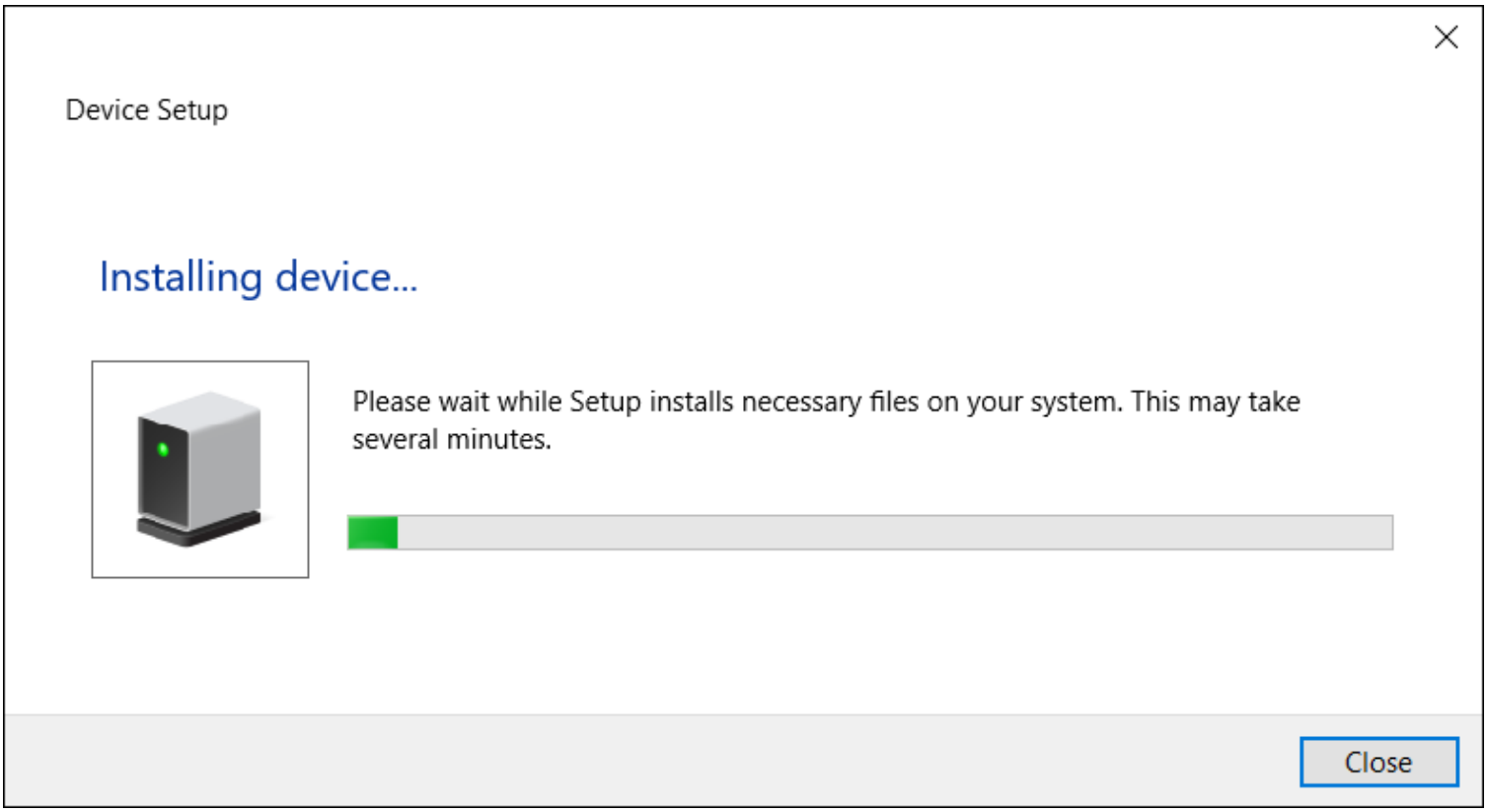

Figure 5-1 Windows Driver Installation

If for some reason, Windows could not automatically locate the driver for you, you might have to install it manually. For FTDI compatible cables, their drivers can be found at FTDI website:

http://www.ftdichip.com/FTDrivers.htm

If the Windows driver is successfully installed, you should be able to find a new COM port in Windows Device Manager, as shown in Figure 5-2.

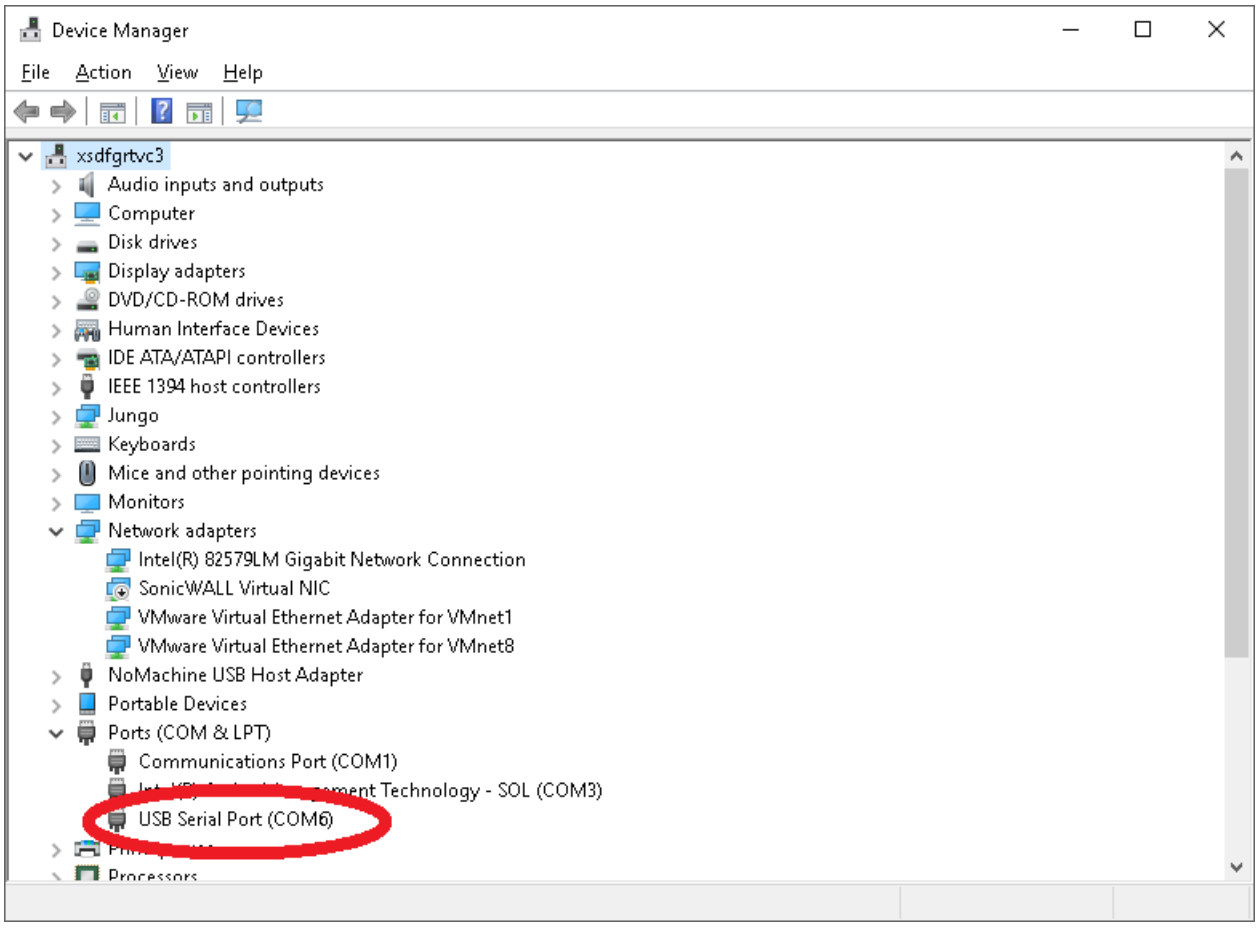

Figure 5-2 USB Serial Port in Windows Device Manager

5.2 Install Arduino IDE

1) After the Windows driver is successfully installed, the next step is to install Arduino IDE. On Windows 10, the Arduino IDE can be installed as an App directly from Windows App store. Otherwise, the Windows installer for Arduino IDE can be found at Arduino Website: https://www.arduino.cc/en/Main/Software

2) After the Arduino IDE is installed, launch it and click the menu File / Preferences, as shown in Figure 5-3:

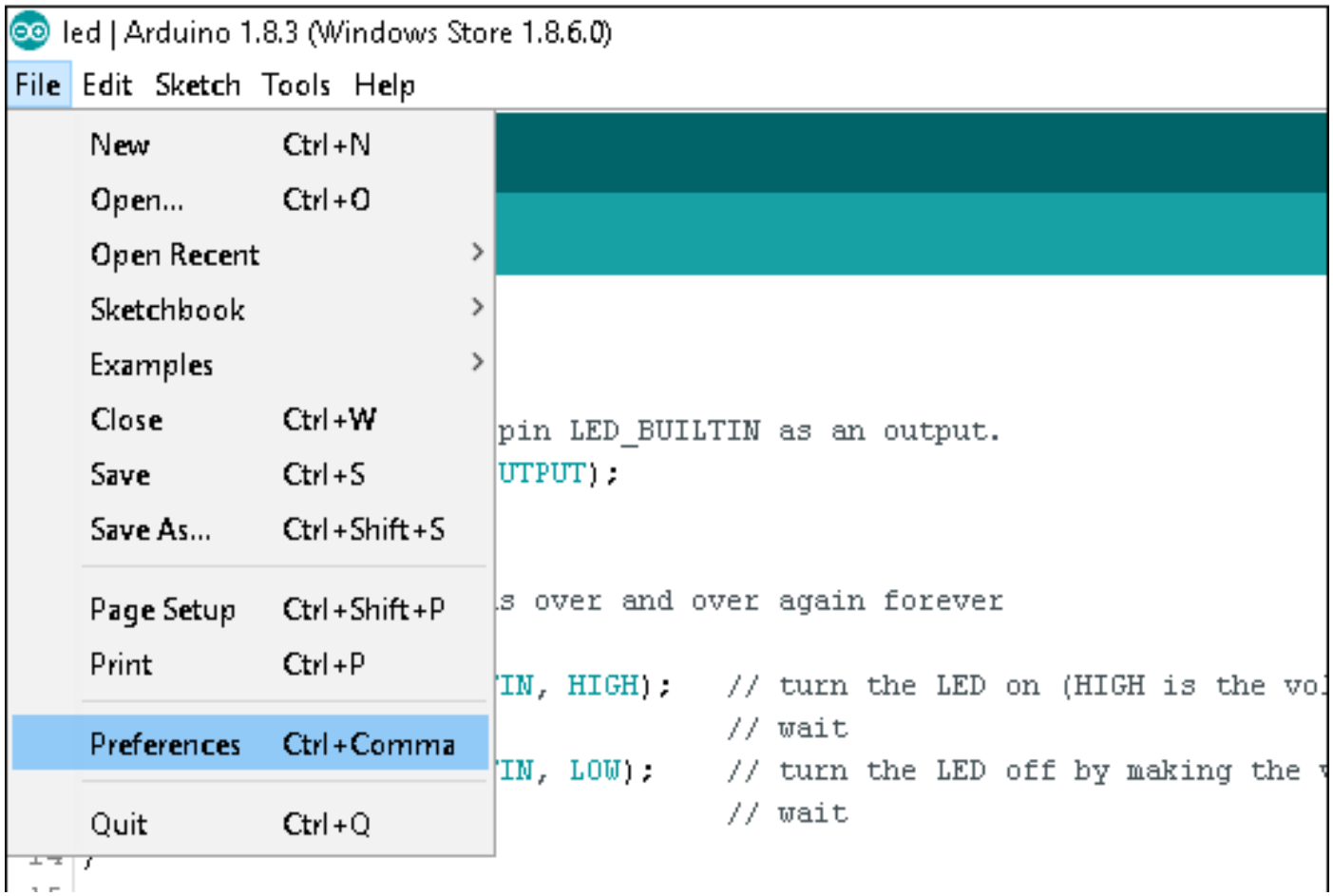

Figure 5-3 Arduino IDE, Preferences Menu

3) The File / Preferences menu will bring out a dialog like the one shown in Figure 5-4. Please set the "Additional Boards Managers URL" to https://raw.githubusercontent.com/PulseRain/Arduino_RISCV_IDE/master/package_pulserain.com_index.json

   (If this input box is not empty, use semicolon to separate multiple URLs.) And click OK to close the dialogue.

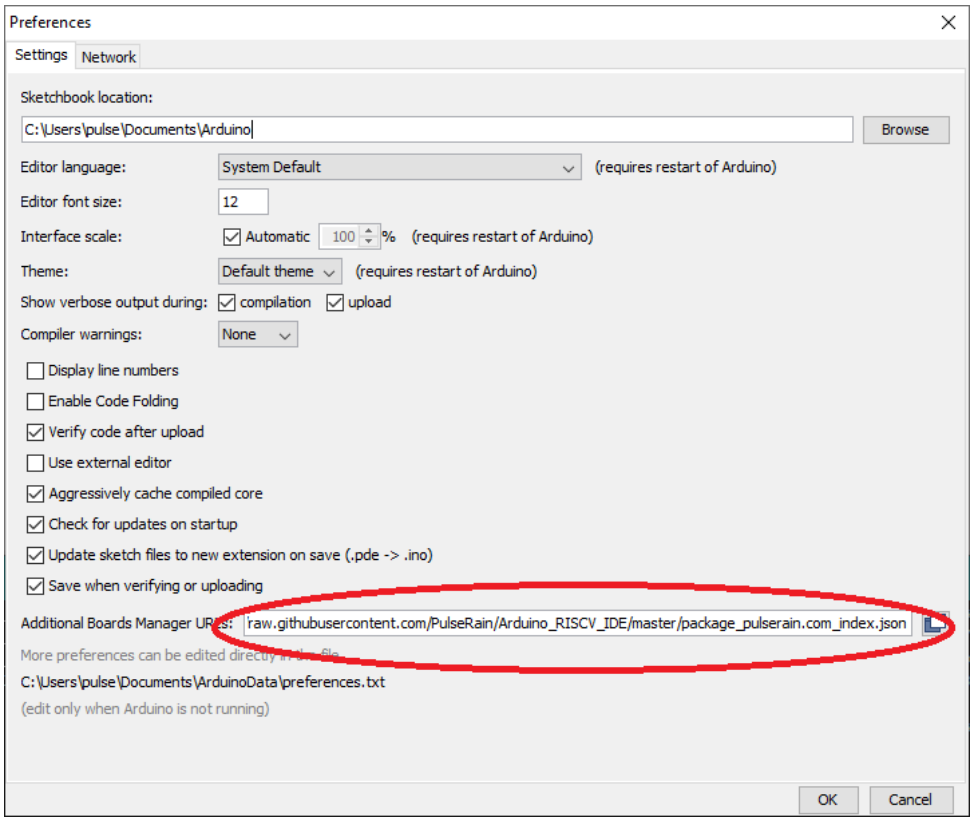

Figure 5-4 Arduino IDE, Preferences Dialogue

4) Now click the menu Tools / Boards / Boards Manager, as shown in Figure 5-5.

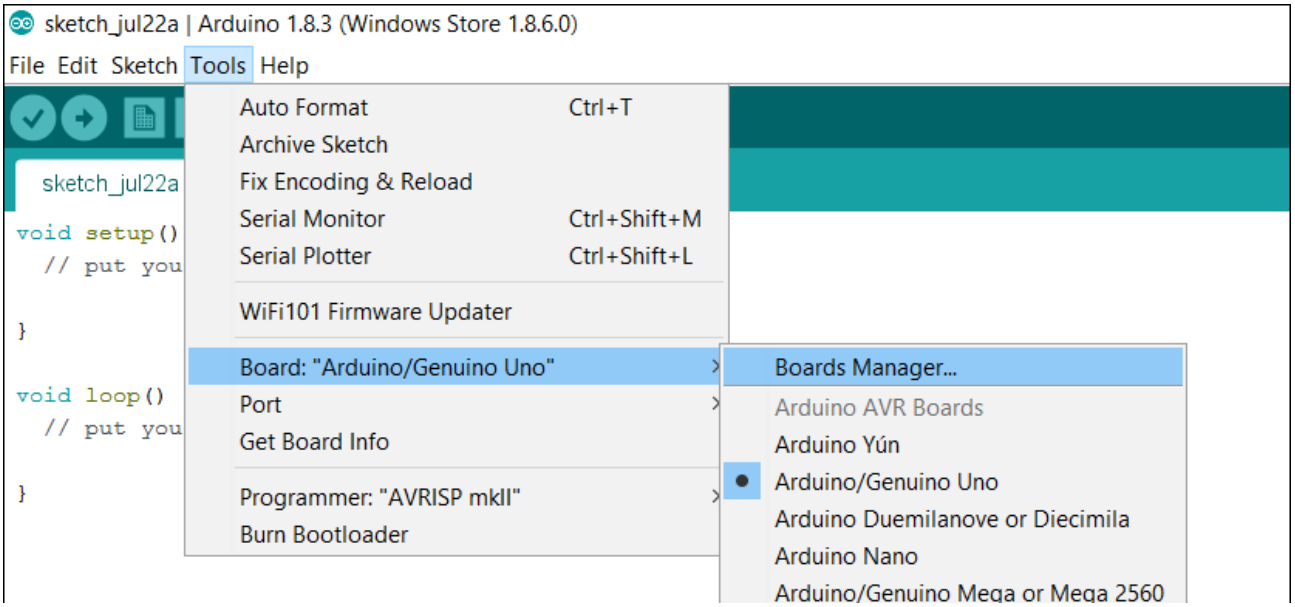

Figure 5-5 Arduino IDE, Boards Manager Menu

5) The "Boards Manager" will bring out a dialogue like the one shown in Figure 5-6. Type in "Reindeer" in the search box to find the board support package for the T20 Dev Kit. Click "Install" to download and install the board support package.

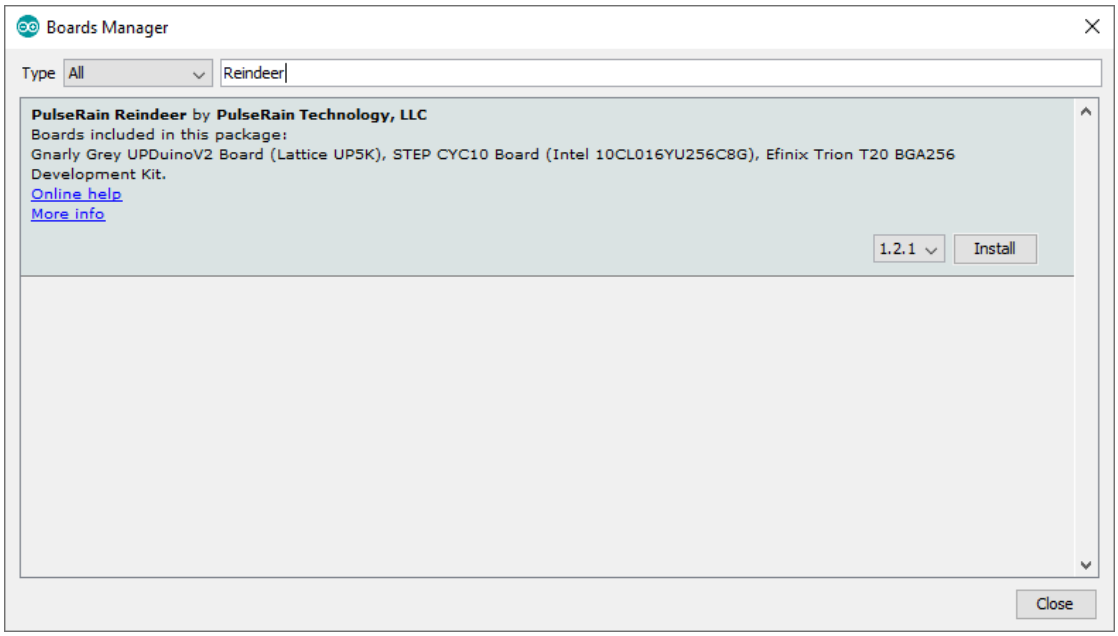

Figure 5-6 Arduino IDE, Boards Support Package for T20 Dev Kit

5.3 Write Sketches

Just like Arduino, you need to select the correct COM port and Board Name before you can start writing sketches.

To select the COM port in Arduino IDE, click the menu Tool / Port, as shown in Figure 5-7. The COM port for the T20 Dev Kit is usually the one that has the biggest index number, but not always. If you have trouble determining which COM port corresponds to the T20 Dev Kit, you can always open the device manager to check, as illustrated in Figure 5-2.

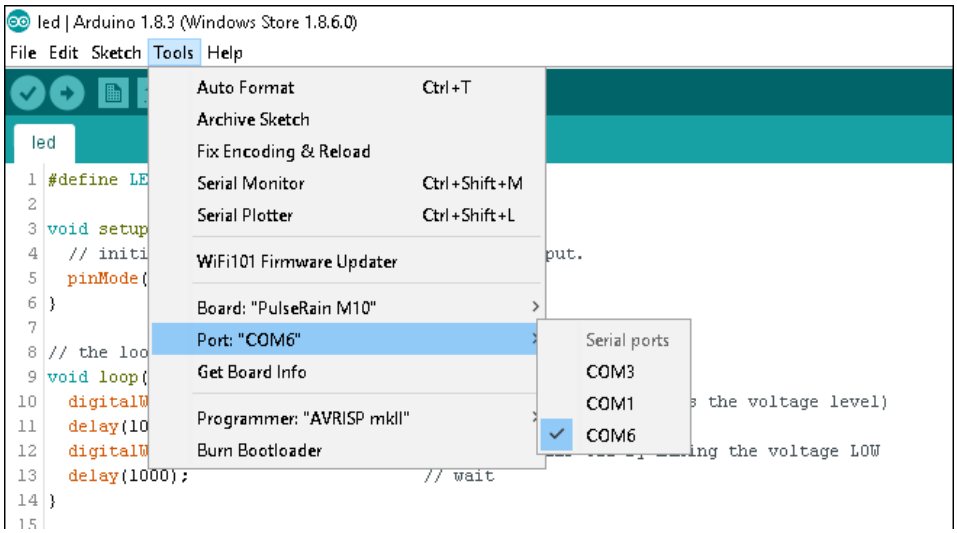

Figure 5-7 Arduino IDE, Select COM port

To select the board name as "Efinix Trion T20 BGA256 Development Kit", click the menu Tool / Board. If the previous steps were done right, you should see the name " Efinix Trion T20 BGA256 Development Kit " somewhere close to the bottom of the menu, as illustrated in Figure 5-8.

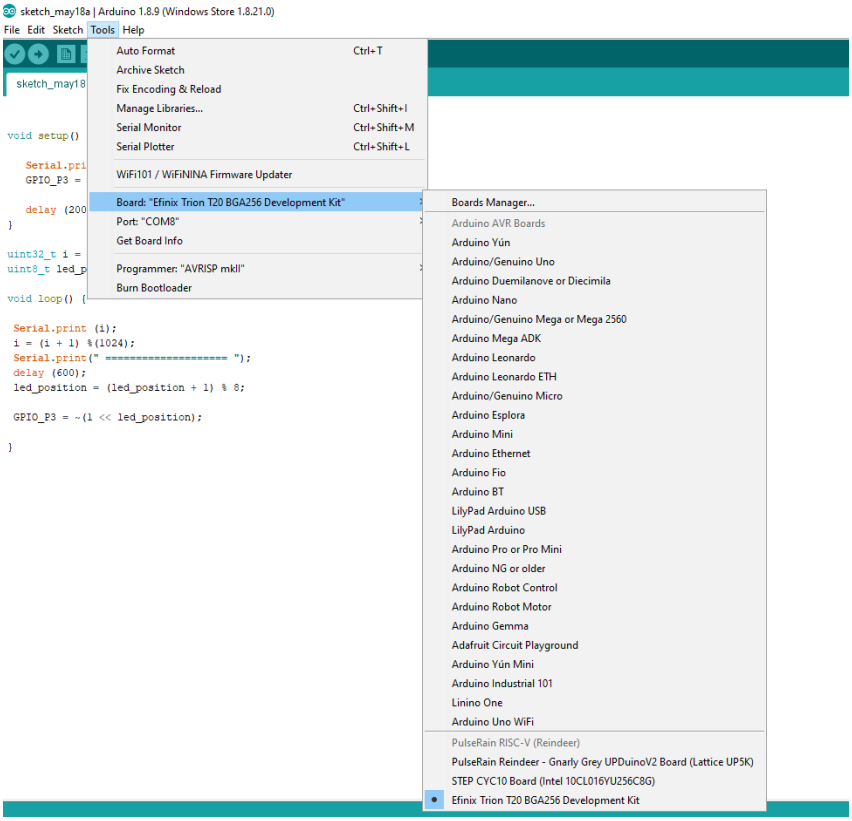

Figure 5-8 Arduino IDE, Select T20 Dev Kit

Now you can start to write some sketches, like the one below test LED and COM Port:

```
void setup() {
    Serial.print(" ====================");
   GPIO_P3 = 0xAA;
   delay (2000);
}
uint32_t i = 0;
uint8_t led_position = 0;
void loop() {
    Serial.print (i);
    i = (i + 1) %(1024);
   Serial.print(" ==================== \n");
   delay (600);
  led_position = (led_position + 1) % 8;
 GPIO_P3 = ~(1 << led_position);
}
```

List 5-1 Sketch to Test LED and COM Port

After you copy and paste the above code into Arduino IDE, you can type in "ctrl-U" (or menu Sketch/Upload) to compile and upload the sketch to T20 Dev Kit. And it is also recommended to turn on the option of "Show Verbose Output" in Preferences Dialogue (Menu File / Preferences), as illustrated in Figure 5-9. In this way, the path of the .hex file can be located through the verbose output.

Also, you can type in “Ctrl-Shift-M” (or menu Tools / Serial Monitor), as illustrated in Figure 5-10, to see the output of the COM port (Please set baud rate to 115, 200)

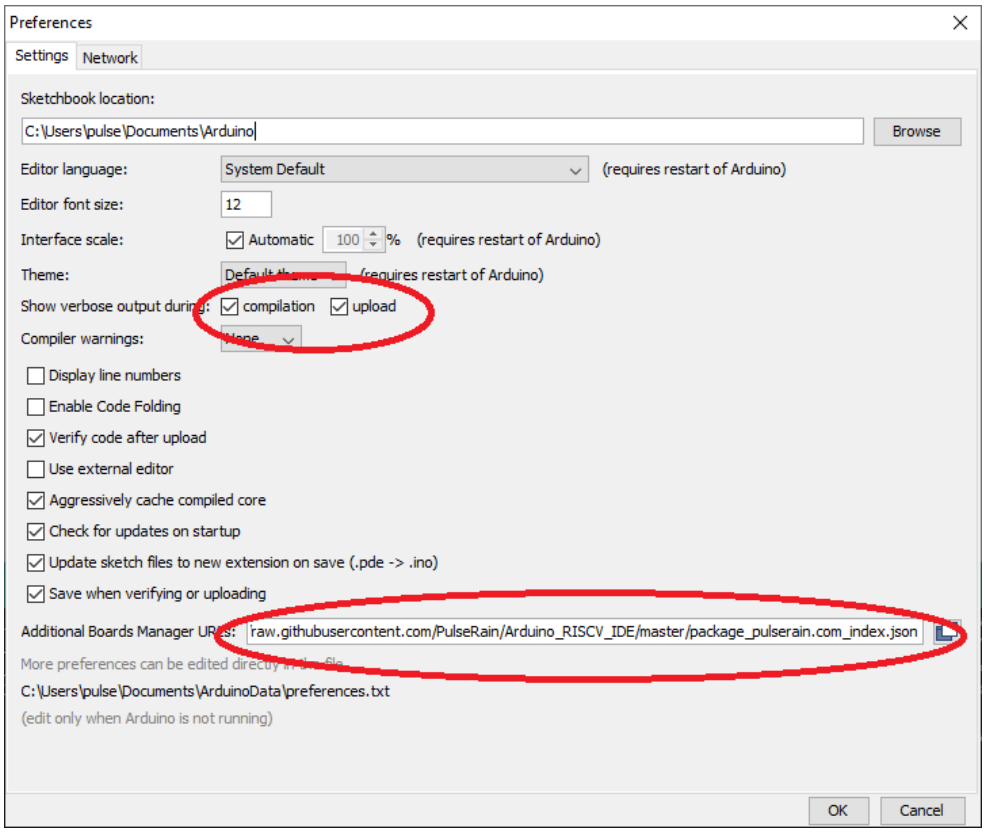

Figure 5-9 Turn on Verbose Option

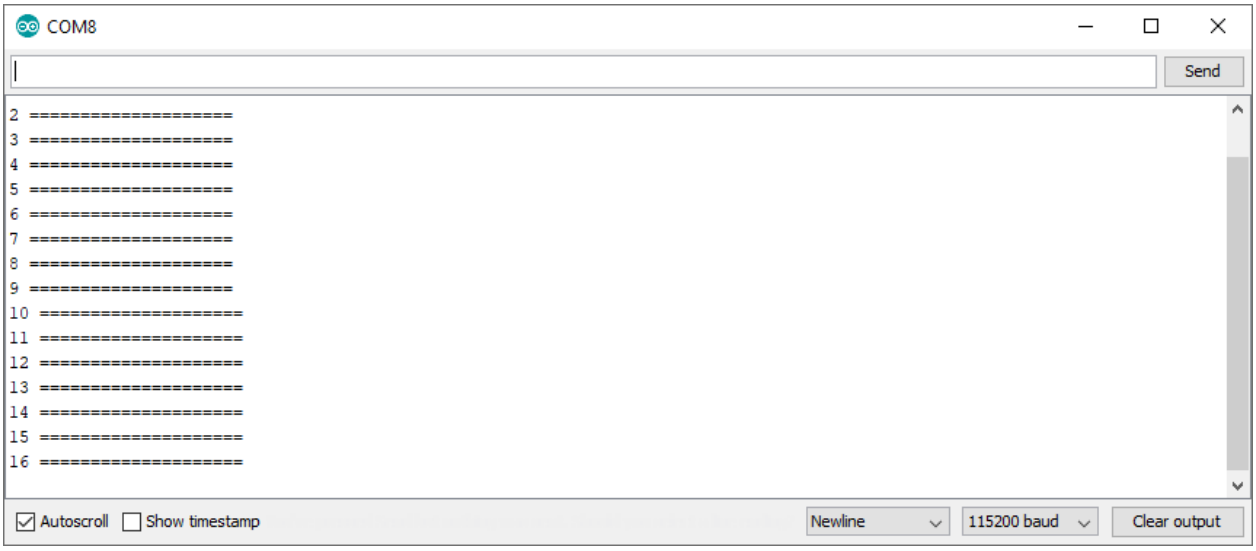

Figure 5-10 Arduino IDE, COM Port output

6 Program the Flash

The T20 Dev Kit is set as SPI Active mode. To program the bitstream into flash, do the following:

1. Connect USB Cable to the onboard microUSB connectorheader (Figure 1-2) and the PC
2. Go to https://zadig.akeo.ie/ and download Zadig tool
3. Launch Zadig, click menu/Options/List All Devices, choose Trion T20 Development Board (Interface 0)
4. Select libusbK and replace Driver (Figure 6-1)
5. Do the same for Trion T20 Development Board (Interface 1)

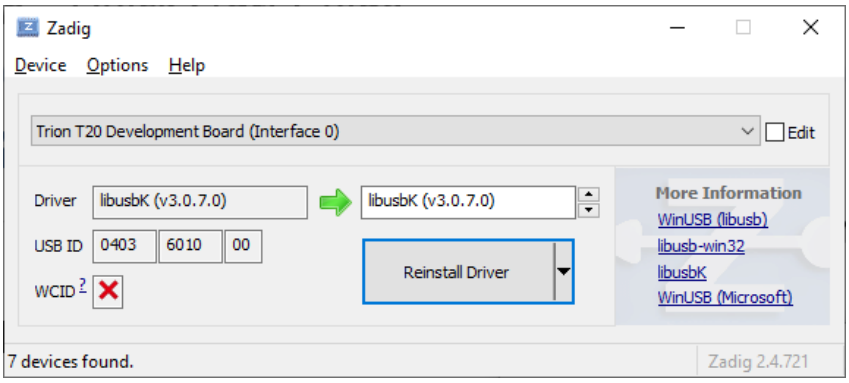

Figure 6-1 Use Zadig to install Driver for T20 Dev Kit

6. Set Power Switch S2 to the "off" position, and then to the "on" position
7. Launch Efinity Programmer, set the USB target to be "Trion T20 Development Board", choose the bitstream file, and set the programming mode to be "SPI Active", as shown in Figure 6-2.

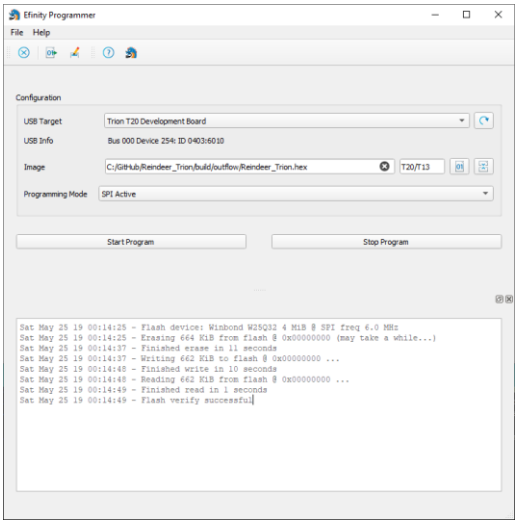

Figure 6-2 Programmer Setup for SPI Flash

8. Click "Start Program"

7 Zephyr

PulseRain Reindeer also supports Zephyr OS. And two Zephyr example applications are provided in https://github.com/PulseRain/Reindeer_Trion/tree/master/scripts/zephyr

- synchronization.elf
- philosophers.elf

Those two Zephyr applications will demonstrate the WFI (wait for interrupt) and semaphore capability.

To download these two .elf files into the FPGA, do the following:

1. Install Python 3.x

2. Install RISC-V toolchain from GNU MCU Eclipse
   https://github.com/gnu-mcu-eclipse/riscv-none-gcc/releases/

3. Make sure Python and the toolchain binaries are accessible by setting up the environment variable of PATH

4. open a command window, and enter the Reindeer_Trion/scripts

5. run "python reindeer_config.py --reset --run --port=COMx --baud=921600 --console_enable --image=zephyr\synchronization.elf" to download the synchronization.elf file to the T20 Dev Kit. The COMx is the COM port assigned by windows for the USB/UART port. And the --image option points to the elf file to be downloaded.

6. If everything goes ok, the user should see the output like the following:

```
C:\GitHub\Reindeer_Trion\scripts>python reindeer_config.py --reset --run --port=COM14 --
baud=921600 --console_enable --image=zephyr\synchronization.elf
================================================================================
# Copyright (c) 2019, PulseRain Technology LLC
# Reindeer Configuration Utility, Version 2.3
================================================================================
baud_rate  =  921600
com_port   =  COM14
toolchain  =  riscv-none-embed-
================================================================================
Reseting CPU ...
Loading  zephyr\synchronization.elf
__start 80000000
```

```
//================================================================
//== Section  vector
//================================================================
    addr = 0x80000000, length = 1044 (0x414)

//================================================================
//== Section  reset
//================================================================
    addr = 0x80004000, length = 4 (0x4)

//================================================================
//== Section  exceptions
//================================================================
    addr = 0x80004004, length = 620 (0x26c)

//================================================================
//== Section  text
//================================================================
    addr = 0x80004270, length = 8560 (0x2170)

//================================================================
//== Section  devconfig
//================================================================
    addr = 0x800063e0, length = 36 (0x24)

//================================================================
//== Section  rodata
//================================================================
    addr = 0x80006404, length = 968 (0x3c8)

//================================================================
//== Section  datas
//================================================================
    addr = 0x800067cc, length = 48 (0x30)

//================================================================
//== Section  initlevel
//================================================================
    addr = 0x800067fc, length = 36 (0x24)
```

```
//================================================================
//== Section  _static_thread_area
//================================================================
      addr = 0x80006820, length = 44 (0x2c)

//================================================================
//== Section  _k_sem_area
//================================================================
      addr = 0x8000684c, length = 32 (0x20)

==================> start the CPU, entry point = 0x80000000

threadB: Hello World from riscv32!
threadA: Hello World from riscv32!
threadB: Hello World from riscv32!
```