Embedded OS 2021

Project assignment

Thomas Herpoel

May 2021

1 Project description

1.1 Goal

The goal of the project is to create a two part system, able to emulate an HID game controller device on one side, and to take input from another kind of HID game controller on the other side. In practice, two STM32F446 micro-controller will be used:

- One of them will be call the **Device emulator**. It will be responsible of emulating a specific kind of USB HID game controller called the **target device**. In order to constrain the project requirements, the target device is going to be an **XBOX 360 game controller**.
- The other MCU will be called the **Translator Host**. As the name implies, it will act as a USB host to handle a **source HID controller** and to translate the reports coming from the source, into the format expected to emulate the target device. In this case, the expected source device will be a **PS3 game controller**.

Figure 1 shows an overview of the system. This will basically allow to use a PS3 controller as if it was a XBOX 360 connected to a PC.

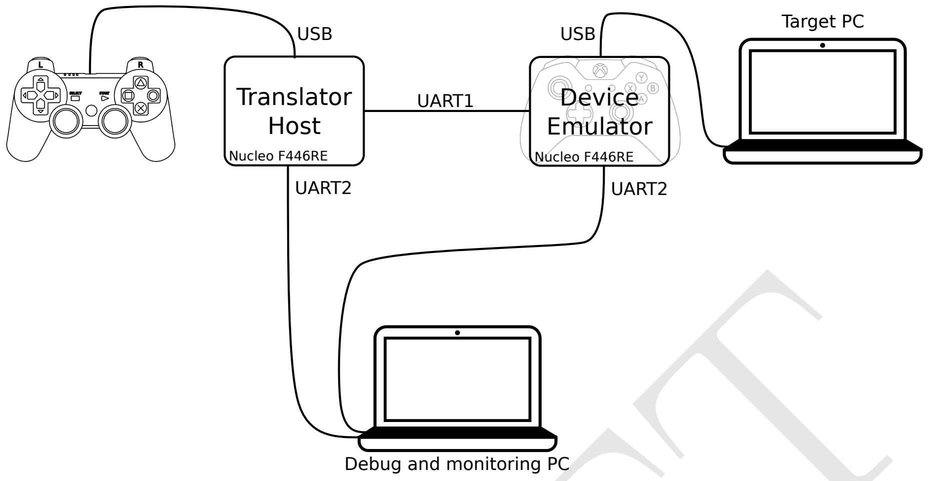

Figure 1: System overview

1.2 Initialization

Figure 2 shows the timing of the system. The first part of the sequence is the USB connection and enumeration:

- Between the device emulator and the target PC, making the target PC believe it is connected to a XBOX 360 game controller. The device emulator then waits to have a message from the translator host telling that the source device was connected, and informing on the source device usb vendor ID and product ID.
- Between the source device and the translator host, letting the translator host knows what is connected to it. It then sends a "Source connected" message to the device emulator, and waits for a "target connected" message, giving also the target usb vendor ID and product ID expected by the target PC. This lets the translator knows what translation to execute between the source device reports and the expected target device reports by the device emulator.

Figure 3 shows what happens if the device emulator is connected afterwards. The translator host just uses a timeout to repeat its "target connected" message.

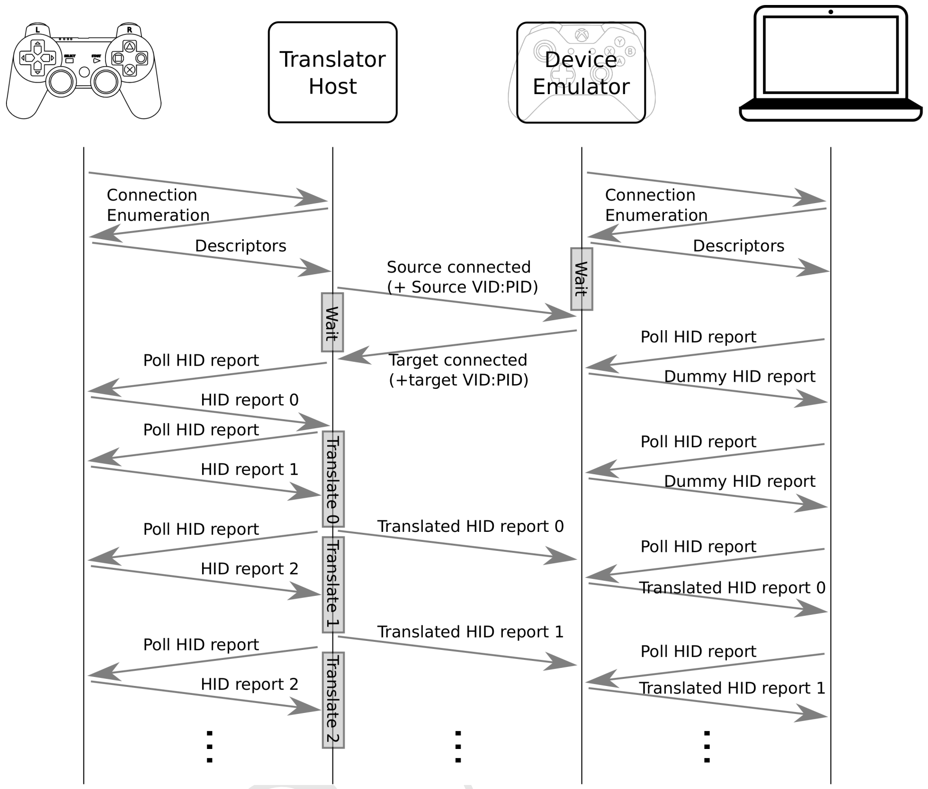

Figure 2: Timing

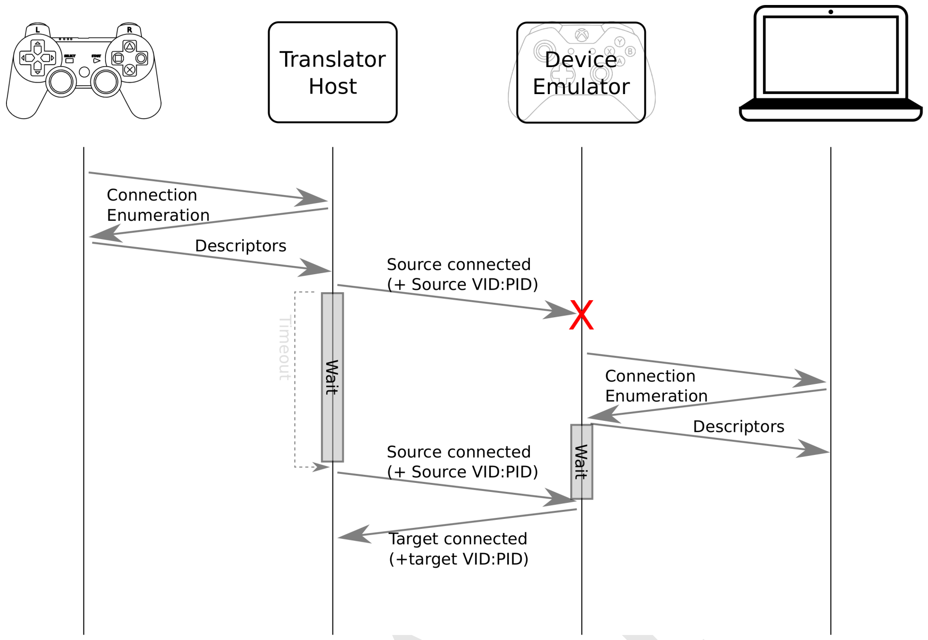

Figure 3: Timeout

1.3 Disconnection

When the source device is disconnected from the translator host, or when the device emulator is disconnected from the target PC, a message will be sent (Source disconnected or Target disconnected) between the two nodes, in order to allow them to gracefully go back to their idle state.

1.4 UART communication between the nodes

The communication between the two parts is handled by a serial UART link. The data will be transmitted in ASCII mode in order to simplify the framing: each frame will be a null terminated string. The messages can have multiple field, separated by a `':'` character. Several messages are possible:

- `SOURCE CONNECTED:[source VID]:[source PID]\0`
- `SOURCE DISCONNECTED\0`
- `TARGET CONNECTED:[target VID]:[target PID]\0`
- `TARGET DISCONNECTED\0`
- `REPORT:[translated HID report]\0`

The VID, PID or translated HID report data will be transmitted in ASCII mode, represented in hexadecimal. For instance, a report message could be:

REPORT:48A7FE510AB675DC...5FBA\0

1.5 State machines

The two parts of the system will use a state machine to operate. They are defined in figures 4 and 5.

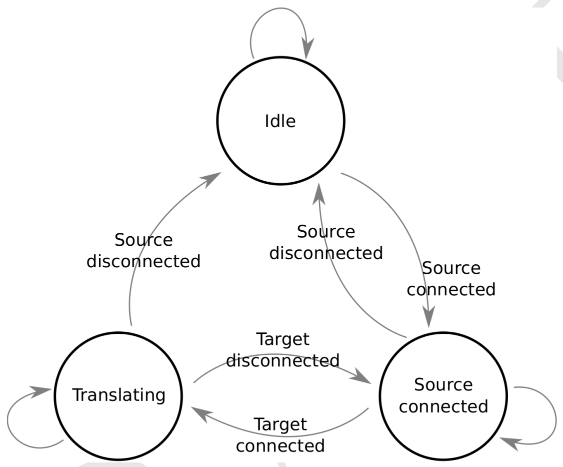

Figure 4: Translator FSM

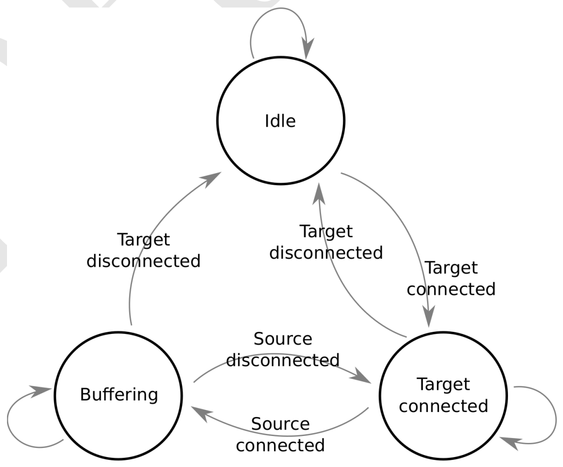

Figure 5: Emulator FSM

2 Assignment

- The class is divided in two teams, each team being responsible of the development of one part of the system (either the translator host or the device emulator).
- Use FreeRTOS **OR** CMSIS-RTOS for **both** part of the project (the teams have to agree on which one to use).
- Use all the mechanisms of real time OSes: protect resources with mutexes, use semaphores, ...
- Don't forget to **comment** your code!
- It is strongly advised to use some versioning tool like **git**!
- Redact a (short) report describing your code, and the modifications added by you to the initial system description if needed.