

Final Project Proposal

Group 21: 陳克盈 (112062205)、蔡明妍 (112062224)

Table of Contents

1	Introduction	2
1.1	Motivation	2
1.2	Overall Introduction	2
2	架構	2
2.1	整體架構	2
2.2	模組細節	3
3	Budget	4
4	Timeline	4

1 Introduction

1.1 Motivation

近期比特幣已經突破九萬美元，市值已經突破 9000 年以來所開採的白銀總值，成為全球第八大資產。雖然距離全就第一大資產黃金還有很大一段距離，但是從近期走勢，以及加密貨幣 ETF 的興起，都顯示加密貨幣已經成為投資者的一大選擇。

近年來高頻交易在金融市場的佔比日益增長，根據研究報導^{*}指出，高頻交易在美國股市的交易量佔比穩定在 50% 以上，而歐洲股市[†]的佔比則在 24 - 43% 左右。

有別於傳統股票市場有著交易所開盤時間的限制，加密貨幣是全球化且全年無休的市場，這也使得加密貨幣成為高頻交易的首選，目前高頻交易佔加密貨幣交易所的交易量穩定在 50%[‡] 以上，其中中國市場更是高達 60 - 80 % 以上[§]。

高頻交易除了一般的程式交易之外，也有不少造市商使用 FPGA 能夠實現硬體加速的特性來進一步地降低計算延遲。雖然我們無法取得 PCIE 等級的 FPGA 板，但我們仍然能夠使用 FPGA 來模擬高頻交易的環境，這也是為什麼我們選擇這個題目作為我們的 Final Project。

1.2 Overall Introduction

在這個專案中，交易將會有兩種模式：

- 手動模式：透過鍵盤輸入，模擬交易員手動下單的情況。
- 自動模式：FPGA 將會內建幾個自動交易策略，FPGA 讀取交易資料後，便會根據策略發出買賣指令。

這兩種模式都將使用 FPGA 作為 IO 與計算的核心，並使用電腦與 FPGA 連接，作為網路封包的傳輸跳板，將對應的 API Request 傳送到 Binance 交易所進行交易。

除此之外，FPGA 板也將會透過 VGA 口，將一個輸入匡顯示到螢幕上，讓使用者能夠知道目前所打的指令。

2 架構

2.1 整體架構

在手動模式下，機器的運作模式主要分成了幾個步驟：

- (1) FPGA 讀取鍵盤輸入，並存入 Buffer 中，並在螢幕上顯示目前的指令
- (2) 當 Enter 鍵被按下時，FPGA 會將 Buffer 中的指令透過設計好的 Binary Protocol 傳到 Host 端
- (3) Host 端的 Decoder 會將 FPGA 傳來的訊號轉換為對應的 API Request，將其傳送到 Binance 交易所後，再經由 Encoder 將交易結果回傳給 FPGA。
- (4) FPGA 收到交易結果後，將相關資訊顯示到螢幕上，並持續監控交易狀況，並根據策略發出額外的交易指令

^{*}<https://www.investopedia.com/terms/h/high-frequency-trading.asp>

[†]https://www.ecb.europa.eu/press/research-publications/resbull/2020/html/ecb.rb201215_210477c6b0.en.html

[‡]<https://blog.ueex.com/cryptocurrency-high-frequency-trading-tactics/>

[§]<https://www.investopedia.com/news/highfrequency-trading-firms-enter-cryptocurrency-markets/>

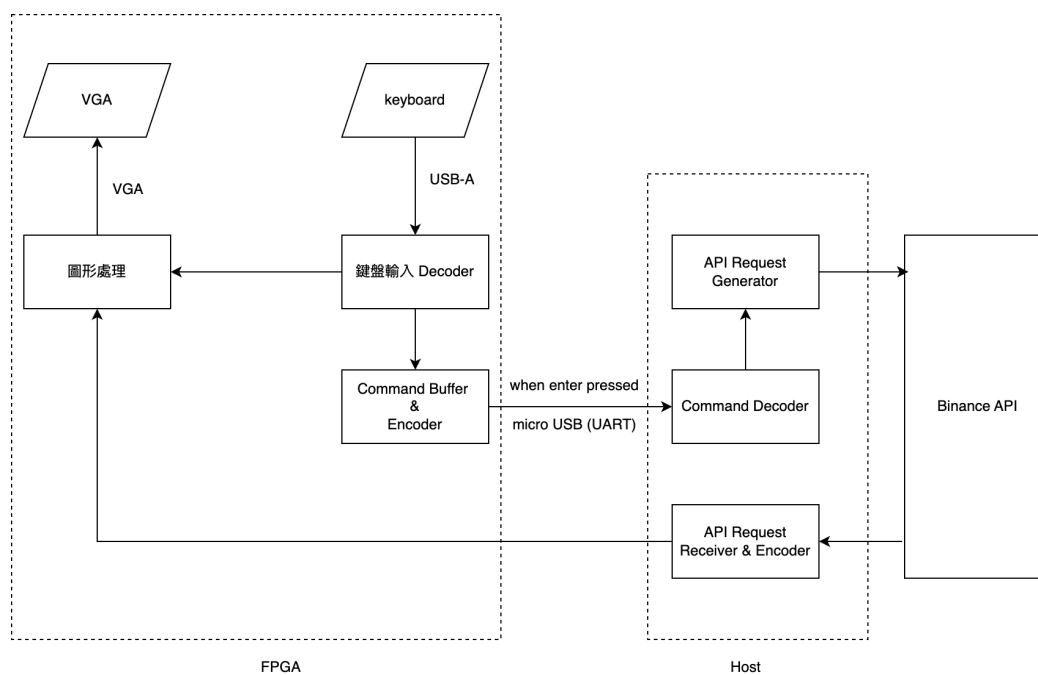


Fig. 1 手動模式架構

在自動模式下，將關閉鍵盤輸入的功能，改由策略模組控制整個交易行為：

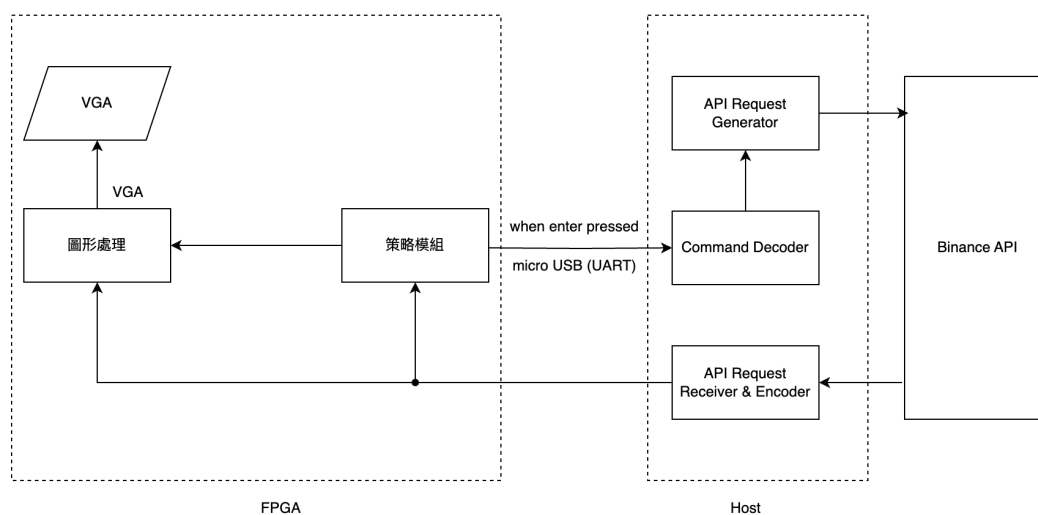


Fig. 2 手動模式架構

2.2 模組細節

鍵盤輸入 Decoder

- 讀取鍵盤輸入
- 支援 Enter, Backspace, Delete, 上下左右等按鍵，讓使用者能夠編輯輸入的內容

- 將輸入內容，以 ASCII Code 的方式儲存到 Buffer 中
- 實時向圖形處理的 module 發送更新過後的輸入內容

圖形處理

將以下資訊轉換為 VGA 訊號顯示到螢幕上：

- 鍵盤輸入內容
- 交易結果
- 帳戶狀態

Command Buffer & Encoder

- 儲存鍵盤輸入的內容
- 當鍵盤輸入 Enter 時，會將 Buffer 內的指令轉換為在自訂 Protocol 中，該指令所對應到的 Binary Code，並將其透過 UART 的方式傳送到 Host 端

Command Decoder & API Request Generator

- 使用 Python 實作
- 接收 FPGA 端傳過來的 Binary Code 後，將其轉換為對應的 API Request
- 將 API Request 透過 Binance API 送到 Binance 交易所

API Request Receiver & Encoder

- 接收 Binance 交易所回傳的交易結果
- 將交易結果轉換為 Binary Code，並透過 UART 傳送到 FPGA 端

策略模組

- 將內建幾種幾單的交易測略，如均線策略、MACD 策略等
- 透過 API Request 來取得目前的價格、成交量資訊，並依據策略送出對應的交易指令

3 Budget

唯一會需要用到錢的只有交易手續費與價差

4 Timeline

- 11/24 - 11/30 : Host 端 API Request
- 12/1 - 12/7 : 鍵盤輸入 & 圖形處理
- 12/8 - 12/14 : Command Buffer / Encoder / Decoder
- 12/15 - demo : 策略模組