

Crypto Monitor on a Real Time Embedded System

Epameinondas Bakoulas

July 2025

1 Objective

Our goal in this project is to implement a crypto monitor that will run on a raspberry-pi continuously. It will collect data via a websocket for a total of 8 different symbols, perform some calculations for each symbol and save the results on .txt files.

Later on, we will expand on this idea by performing other calculations out of the scope of this project, in order to better predict the market and make better decisions on when to buy or sell.

2 Analysis

2.1 Overview

Our analysis will be performed on 8 different cryptocurrency symbols, which are BTC-USDT, ADA-USDT, ETH-USDT, DOGE-USDT, XRP-USDT, SOL-USDT, LTC-USDT, BNB-USDT, and we will use the the OKX Websocket public channel to collect the exchange data.

We're going to perform a total of 2 calculations for each symbol.

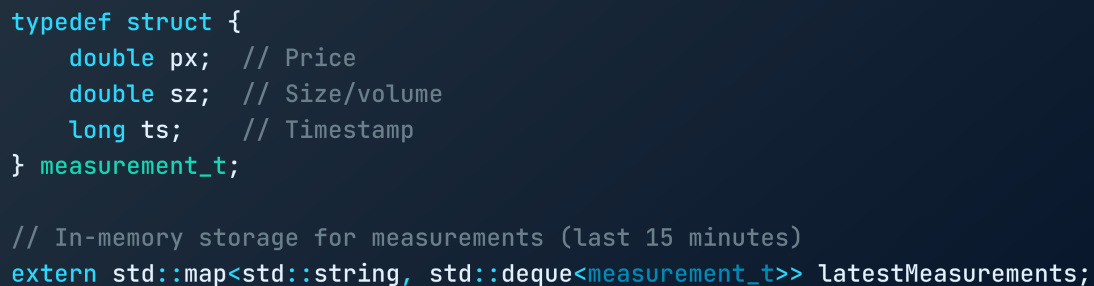
- Calculate the moving average price and volume of the symbol for the last 15 minutes.
- Calculate the pearson correlation with all other symbols (and itself) using the last 8 averages (we will clarify on this later on).

These calculations are going to be performed every minute and the results will be saved on a .txt file.

2.2 Storing measurements

The measurements (exchanges) that arrive via the websocket will be saved in 2 places:

- In-memory for quick retrieval and calculations (last 15 minutes only), on a double-ended queue.
- In .txt files for long-term storage and analysis, separate for each symbol.



```
typedef struct {
    double px; // Price
    double sz; // Size/volume
    long ts;   // Timestamp
} measurement_t;

// In-memory storage for measurements (last 15 minutes)
extern std::map<std::string, std::deque<measurement_t>> latestMeasurements;
```

Figure 1: Measurement Storage

2.3 Scheduling

In order to perform the calculations every minute, we will spawn a thread called `threadScheduler` that will run every minute and perform the calculations.

3 Benchmarks

4 Conclusion

The source code can be found on the Github repository.