# Getting Started with Development

## Development Environment Setup

- IDE: Visual Studio Community
- SDK: SON Library?
  - https://ced.co.uk/upgrades/spike2tools
- CanUSB:
  - https://www.canusb.com/support/canusb-support/
- Additional Libraries
  - Thread management (Intel TBB? `vcpkg install tbb:x64-windows`)
  - Serial Communication (Boost? `vcpkg install boost-asio:x64-windows`)

## Possible Project Configuration

```
ProjectRoot/
├── src/
│   ├── can_monitor/
│   ├── spike2_integration/
│   └── threading/
├── include/
├── lib/
├── tests/
└── docs/
```

## Relevant sections from the documentation

Harmonic Drive CANopen Manual: Focus on sections related to Cyclic Synchronous Position Mode (CSP, Chapter 9). Specifically:

- Object 6040h (Control Word): Used to control the drive's state machine (start, stop, enable, etc.).
- Object 6041h (Status Word): Provides feedback on the drive's status (e.g., operation enabled, fault, target reached).
- Object 6064h (Position Actual Value): This is the primary data you'll be reading – the actual position of the motor.
- Object 607Ah (Target Position): While you won't be setting this in your application (Spike2 will handle target positions), understanding this object is helpful for context.
- Object 60C2h (Interpolation Time Period): Important for understanding the timing of position updates.

CANUSB Manual: Concentrate on sections about:

- Installation (Chapter 1.2): Make sure you have the correct drivers (D2XX recommended for real-time performance).
- Testing (Chapter 1.3): Basic verification of communication.
- Available Commands (Chapter 2): You'll need commands like 'O' (Open CAN channel), 'S' (Set bitrate), 't' or 'T' (Transmit CAN frame), and 'r' or 'R' (Request CAN frame). The 'F' command (Read Status Flags) can

be used for error handling.

## Requirements/Notes

Timing accuracy: 1000 Hz - ms by ms max total latency - 499 microseconds?

## Questions & Considerations

What version of Spike2 (8, 9, 10? 32 bit or 64 bit) https://ced.co.uk/downloads/latestsoftware

interpolation or extrapolation methods for time syncing

Error checking for data loss