1 Preface

This manual is written with the explict intent to fast-track a new member through software training and get them working with real embedded platforms as soon as possible. However, this manual may also serve as a guide for anyone as a simple introduction to embedded programming. This manual is written with zero expectations of the reader having any previous experience in embedded programming, or even programming in general. While it may be beneficial to have prior experience, it is absolutely not necessary to pocess any knowledge beyond common sense before reading this guide.

There are obvious things missing if you want to use this as a software programming guide. For instance, advanced topics that are unhelpful for embedded programming, like OOP (object-oriented programming), pointers operations, data types (especially non-integer types like floats and strings), data structures (stack, linked-list (this is garbage anyway), graph, tree) have either been omitted or shorted significantly. For instance, there is no benefit for having inheritance in the VCU when all you are handling are CAN and a few analog inputs.

There are also oddly-specific topics that is delved deep into. This is mostly out of actual demands. For instance, interrupts and communication protocols form a significant part of embedded programming, and is therefore heavily emphasized.

This manual was written to complement the 2025 Red Bird Racing Software training and therefore contains exercises that are intended for a teacher to mark. While these exercises would still be included in this manual alongside suggested answers, it is best to have a teacher reading through the exercise to see the strong and weak points of the student.

Although this manual is written with the avr platform, and more specifically the atmega328p platform in mind, the concepts and practices should still be applicable to other platforms like ESP32 or STM32. However, like emphasized in this manual, the reader is reminded to read the documentation of the platform they are developing, especially for lower-level operations like register manipulation and interrupts, as well as for how to set up their development platform.