Lessons: 00.Arduino-examples

An introduction to Cosa

Version: 0.0.2, Date: 2016-10-20

Contents

- General Differences

- Programming Style

- Performance and Foot-Print

- Detailed Differences

- Overall Architecture

- Lessons

General Differences

- Cosa is object-oriented replacement for the Arduino core.

- It can be used with the Arduino IDE where it installs itself as a new core with supported boards.

- Cosa does not implement the Arduino API. Instead it is an object-oriented platform for programming AVR based boards such as Arduino.

Programming Style

- Cosa has a very specific programming style with well documented and integrated classes.

- It follows much of the style found in industry.

- A fully hyper-linked and graphical documentation can be generated from the source code with Doxygen.

- Each release of Cosa is bundled with both source and documentation. The documentation is also available on-line.

Performance and Foot-Print

- Cosa is a large set of integrated classes that reflect the AVR hardware modules and functionality.

- The design is both faster and uses a lower foot-print than the Arduino core.

- The object-oriented design (OOD) uses C++ language features to achieve a rich API but at the same time avoid exhaustive dynamic memory usage.

- Heap allocation is avoided throughout the design instead static objects/instances are used together with stack based allocation.

Detailed Differences

- An Arduino sketch implicitly includes Arduino.h and generates forward declarations of functions in the sketch. This is done to make it easy for newcomers.

- A Cosa sketch must explicitly include an components (header files) used.

- The sketch is also responsible for calling initialization code for the components.

- As Cosa is designed to support low-power mode hardware modules such as ADC, Timers, etc, are not started automatically.

Overall Architecture

- Cosa uses a number of design patterns to support both sketches and extensions such as device drivers.

- A sketch typically uses an abstraction of a hardware function while a device driver mediates the functionality.

- An example is the abstraction of IOStream which is used for both internal buffers, serial communication, networks, displays, widgets, etc. An IOStream device driver must implement a specific set of virtual member functions.

- This architecture is often called North- and South-bound Interface where North-bound is the sketch/application interface and South-bound is the extension interface.

- The design pattern is known as delegation.

Lessons

- This set of lessons introduce Cosa by rewriting of the Arduino examples.

- Each sketch contains a description and circuit.

- It is recommended to display the Cosa and Arduino sketches side-by-side in the Arduino IDE.

- There may be several variants of the same Arduino example sketch to demonstration of different aspects of Cosa.