

Dartino: Modern Embedded Programming

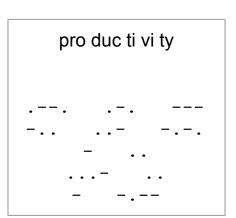
Product summary, February 2016

### Introducing Dartino

- Experiencing vast growth in embedded devices & IoT?
- Need to quickly prototype, customize, and productize?
- Creating graphical user interfaces and/or secure communication over BLE and WiFi?

Writing application code using C/C++ is costly and error prone.

Dartino enables a modern programming language on embedded microcontrollers, reducing development cost by saving time and improving quality and security.





# Modern Programming for Embedded

- Program embedded with a modern, high-level language, Dart
- Elegant and familiar syntax
- Easy to learn based on current experience from C, C++, Java, JavaScript, Python, C#, etc.
- Optional typing: Agile prototyping yet strong validation
- Better testing: Easy to run and automate code on a PC

```
main() {
  // Turn on the status LED
  const int statusLed = 4;
  var gpio = new GPIO();
  gpio.setMode(statusLed, Mode.output);
  gpio.setPin(statusLed, true);
  // Send current temperature as a MQTT message
  Client c = new Client(
    'tcp://test.mosquitto.org:1883', 'test-client');
  c.publish('Temp: $currentTemp', '/sensors/temp');
  c.disconnect();
```



## Interoperability with existing C/C++ and native code

- Combine Dartino code with existing code enabling reuse
- Generate Dartino wrapper libraries automatically from header files [IN PROGRESS]

```
// Initialize a library from a shared library file (.so/.dll)
final ForeignLibrary mqttLib = new ForeignLibrary.fromName(
    ForeignLibrary.bundleLibraryName('paho-mgtt3c'));
// Call function in native code and return result to Dartino code
ForeignMemory mgttClientHandle = new ForeignMemory.allocatedFinalized(8);
int result = mqttLib.lookup('MQTTClient_create').icall$5(
    mqttClientHandle,
    new ForeignMemory.fromStringAsUTF8(serverURI),
    new ForeignMemory.fromStringAsUTF8(clientID),
    ForeignPointer.NULL);
```



## Free & Open Source

- Dartino is free to use. This includes both developer tools & the runtime
- Dartino is developed as a full open source project on GitHub
  - Transparent development process
  - Community participation encouraged
  - Endless customization opportunities
- Integrates with other open source
  - Embedded OS: FreeRTOS <u>link</u>
  - Sockets & TCP/IP: FreeRTOS+TCP link
  - o SSL/TLS: mbed TLS link



https://github.com/dartino/



### Hardware platform support

- Dartino code can be run both locally on the developer
   PC, and on a attached embedded device
- Runtime is available on ARM Cortex processors
  - Medium to large micro-controllers (MCUs):
    - Cortex M3, M4, and M7 processors
    - 512kb flash or more
    - 128kb RAM or more
  - Micro-processors (MPUs)
    - Raspberry Pi 2
- Integrates with developer and custom boards using FreeRTOS and silicon vendor drivers and HALs





#### Tools

- Free open source tools
  - Command line compilation, image building, and flashing
  - IDE with syntax
     highlighting, go to
     declaration, etc.
  - Debugger
     (breakpoints, stack, view variables, etc.)
     [IN PROGRESS]

```
000
                                                               main.dart — /Users/mit/tmp/flutter/sweet_river_8005
 flutter1
                                                                                                                    Debugging lib/main.dart
                                  import 'package:flutter/material.dart';
 sweet river 8005
  > atom
                                  void main() {
                                                                                                                    debug drawing slower animations
  dua. 🔳 <
                                    runApp (
  > apk
                                                                                                                    Isolate dev.flx$main (paused)
                                       new MaterialApp(
  ▼ 🛅 lib
                                         title: "Flutter Demo",
      main.dart
                                         routes: <String, RouteBuilder>{
                                                                                                                    EXECUTION LIBRARIES ISOLATES
  > packages
                                           '/': (RouteArguments args) => new FlutterDemo()

    ≡ FlutterDemoState.incrementCounter.<anon>()

    gitignore.
    packages.
    flutter.yaml
                                                                                                                    pubspec.lock
                                                                                                                    ■ _InkResponseState._handleTap() package:flutter/src/material/ink_... #10
    pubspec.yaml
                                                                                                                     ▶ this FlutterDemoState
                                  class FlutterDemo extends StatefulComponent {
    README.md
                                    State createState() => new FlutterDemoState();
                                                                                                                    BREAKPOINTS
                                  class FlutterDemoState extends State {
                                    int counter = 0;
                                                                                                                    sweet_river_8005 lib/main.dart line 8
                                                                                                                     sweet river 8005 lib/main.dart line 16
                                    void incrementCounter() {
                                                                                                                    sweet_river_8005 lib/main.dart line 24
                                                                                                                    flutter1 lib/main.dart line 25
                           24 →
```



#### Additional info and contact details

Email (Dartino Product Manager):

mit@google.com

Homepage [IN PROGRESS]:

https://dartino.org

GitHub repo:

https://github.com/dartino/sdk

