Flutter in Embedded

Andrea Ricchi
Amarula Solutions
Fosdem 2024

About Us

Andrea Ricchi
andrea.ricchi@amarulasolutions.com
https://github.com/AndreaRicchi

Embedded Linux consultant & developer at **Amarula Solutions**.

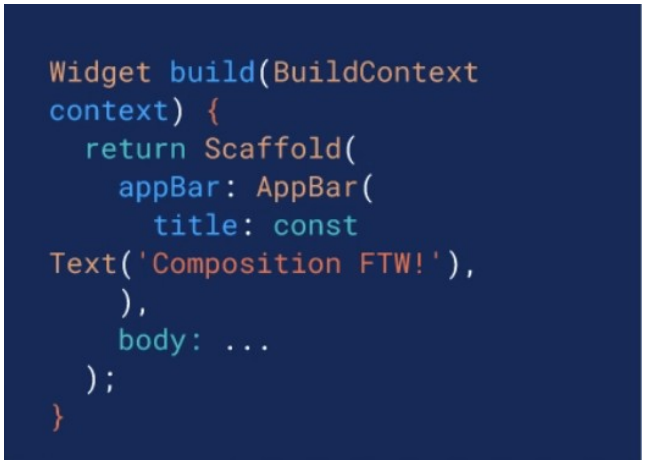
C++ and UI expert with different frameworks.

What is this about?

This talk will guide you through the usage of Flutter as UI framework in Embedded projects, going from the build system integration, development and final result.

What is Flutter?

Flutter is an open source framework by Google for building beautiful, natively compiled, multi-platform applications from a single codebase.
It uses Dart as programming language.



Why Flutter?

Flutter is **fast**

Flutter code compiles to ARM or Intel machine code as well as JavaScript, for fast performance on any device.
You can expect excellent performance. Flutter is designed to help developers easily achieve a constant 60fps. Flutter apps run using natively compiled code—no interpreters are involved. This means that Flutter apps start quickly.

```
class MyApp extends StatelessWidget
{
  const MyApp({Key? key}) :
super(key: key);
  @override
  Widget build(BuildContext
context) {
    return MaterialApp(
      theme: ThemeData.light(),
      home: const CounterPage(),
    );
  }
}
```

Why Flutter?

Dart is **optimized for UI**

Mature and complete async-await for user interfaces containing event-driven code, paired with isolate-based concurrency
A programming language optimized for building user interfaces with features such as sound null safety, the spread operator for expanding collections, and collection if for customizing UI for each platform. A programming language that is easy to learn, with a familiar syntax

```
fetchTemperature() async {
  // This call is non-blocking
  // The UI thread will continue
  // to render with no locks
  final response = await
    http.get('https://my/weather');

  if (response.statusCode == 200) {
    temp.setText(response.body);
  } else {
    temp.setText('Unknown temp.');
  }
}
```

Why Flutter?

Flutter is **productive**

Make changes to your source code iteratively, using hot reload to instantly see the effect in the running app
Write code using a flexible type system with rich static analysis and powerful, configurable tooling. Do profiling, logging, and debugging with your code editor of choice.

Why Flutter?

Flutter is **flexible**

Control every pixel to create customized, adaptive designs that look and feel great
on any screen.

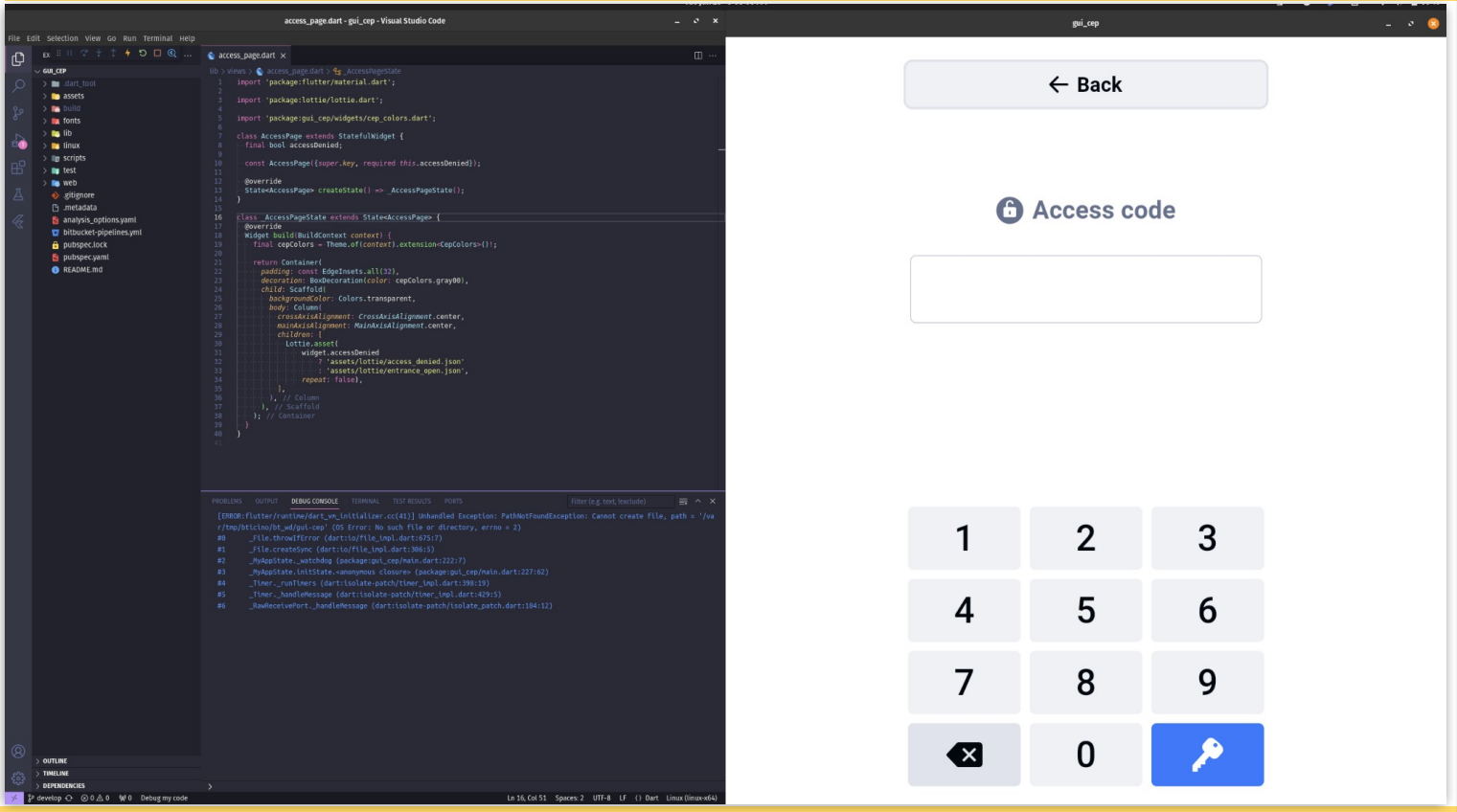
```
Scaffold(
  appBar: AppBar(
    title: const Text('Automatic adaptivity!'),
  ),
  body: ...
)
```

AOT-compile apps to native machine code for instant startup. Target the web with complete, mature, fast compilers for JavaScript. Run backend code supporting your app, written using a single programming language

```
$ dart compile exe hello.dart
$ time ./hello.exe
Hello, Developer!

Real 0m0.012s
# Instant startup; completed in just 12ms
$
```

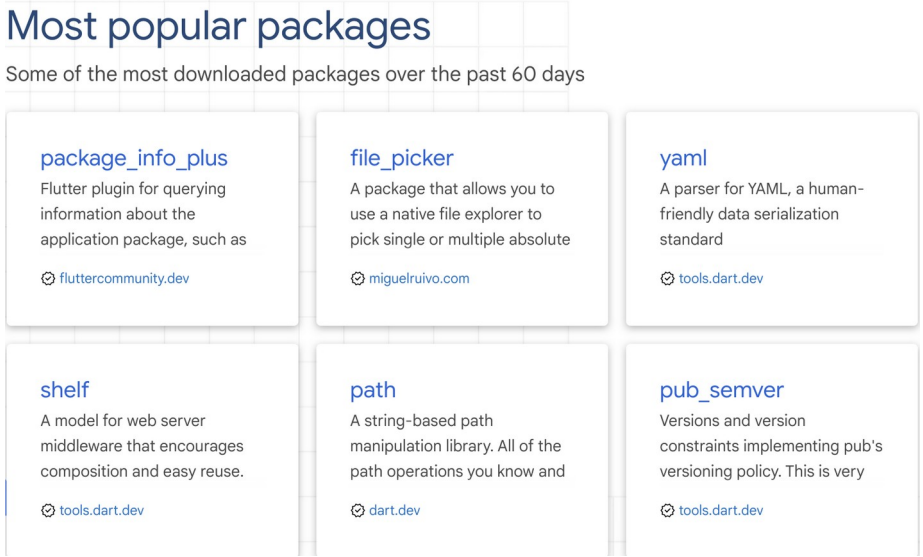
IDE Setup


access_page.dart - gui_cep - Visual Studio Code
File Edit Selection View Go Run Terminal Help
GUI_CEP
access_page.dart
import 'package:flutter/material.dart';
import 'package:lottie/lottie.dart';
import 'package:gui_cep/widgets/cep_colors.dart';
class AccessPage extends StatefulWidget {
final bool accessDenied;
const AccessPage({super.key, required this.accessDenied});
@override
State<AccessPage> createState() => _AccessPageState();
}
class _AccessPageState extends State<AccessPage> {
@override
Widget build(BuildContext context) {
final cepColors = Theme.of(context).extension<CepColors>()!;
return Container(
padding: const EdgeInsets.all(32),
decoration: BoxDecoration(color: cepColors.gray00),
child: Scaffold(
backgroundColor: Colors.transparent,
body: Column(
crossAxisAlignment: CrossAxisAlignment.center,
mainAxisAlignment: MainAxisAlignment.center,
children: [
Lottie.asset(
widget.accessDenied
? 'assets/lottie/access_denied.json'
: 'assets/lottie/entrance_open.json',
repeat: false),
],
), // Column
), // Scaffold
); // Container
}
}
[ERROR:flutter/runtime/dart_vm_initializer.cc(41)] Unhandled Exception: PathNotFoundException: Cannot create file, path = '/var/tmp/bticino/bt_wd/gui-cep' (OS Error: No such file or directory, errno = 2)
#0 _File.throwIfError (dart:io/file_impl.dart:675:7)
#1 _File.createSync (dart:io/file_impl.dart:306:5)
#2 _MyAppState._watchdog (package:gui_cep/main.dart:222:7)
#3 _MyAppState.initState.<anonymous closure> (package:gui_cep/main.dart:227:62)
#4 _Timer._runTimers (dart:isolate-patch/timer_impl.dart:398:19)
#5 _Timer._handleMessage (dart:isolate-patch/timer_impl.dart:429:5)
#6 _RawReceivePort._handleMessage (dart:isolate-patch/isolate_patch.dart:184:12)
OUTLINE
TIMELINE
DEPENDENCIES
develop
Debug my code
Ln 16, Col 51 Spaces: 2 UTF-8 LF Dart Linux (linux-x64)
gui_cep
Back
Access code
1
2
3
4
5
6
7
8
9
0

Why Flutter?

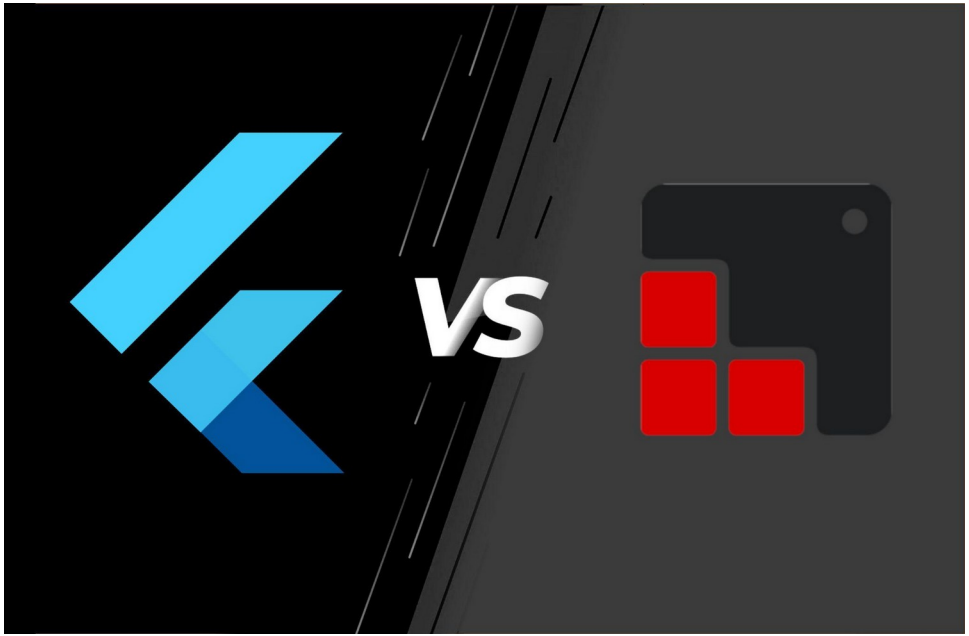
Flutter starting from mobile to embedded is become very popular with a big community and interest from companies.

- Active community on different platforms
- Huge list of packages
- Actively developed and updated
- Used by big tech companies:
  - Google
  - BMW
  - Toyota



Flutter vs LVGL

- C vs Dart
- Hot reload and hot restart against rebuild and redeploy
- More platform supported
- More available packages (libs)
- Bigger community
- Easier to build and publish



Flutter vs Qt

- C++/QML vs Dart
- Hot reload and hot restart against rebuild and redeploy
- Free and open source against commercial license
- Rapidly improving

When things get complicated

Getting complex with **FFI**

With foreign function interface (FFI), Dart is capable of to call native C APIs, and to read, write, allocate, and deallocate native memory.

Embedded integration

Flutter package is available both for Yocto and Buildroot.

- Yocto: meta-flutter on GitHub is the official Flutter layer
- Buildroot: Flutter package maintained by my co-worker Adam Duskett

Product example

Q&A